

**PENGURANGAN NOISE PADA CITRA DIGITAL
MENGUNAKAN METODE STATISTIK MEAN, MEDIAN,
KOMBINASI DAN REKURSIF FILTER**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

Oleh :

M. HAKIKI OCEANDRA
10651004381



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2013**

PENGURANGAN NOISE PADA CITRA DIGITAL MENGUNAKAN METODE STATISTIK MEAN, MEDIAN, KOMBINASI DAN REKURSIF FILTER

M. HAKIKI OCEANDRA
10651004381

Tanggal Sidang : 24 Juni 2013
Periode Wisuda : November 2013

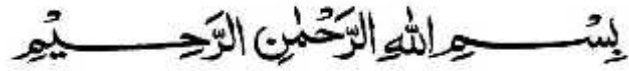
Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Perkembangan teknologi pencitraan yang menghasilkan citra digital masih meninggalkan permasalahan kualitas yang menurun akibat noise. Penelitian ini mempelajari teknik-teknik pengurangan noise dengan menggunakan metode statistik berbasis mean dan median. Pengembangan basis filter secara rekursif dan kombinasi untuk mengurangi noise selanjutnya dilakukan, dan diuji coba terhadap beberapa citra *benchmark*, yaitu Lena dan Baboon. Citra uji diberikan noise berupa impulse Salt & Pepper dan additive Gaussian, kemudian dianalisa performanya secara kualitatif dengan membandingkan citra output filter, citra bernoise dan citra asli secara kasat mata. Secara kuantitatif diukur melalui nilai PSNR-nya. Pengujian lebih mendalam terhadap performa filter rekursif dilakukan terhadap citra pas foto. Hasil pengujian dapat disimpulkan karakteristik noise impulse dapat dikurangi secara signifikan dengan menggunakan filter median, atau filter rekursif dan kombinasinya, yang ditunjukkan dengan nilai PSNR yang tinggi di atas 17dB untuk level noise yang tinggi (50%). Sedangkan noise Gaussian dapat dikurangi menggunakan filter mean, rekursif atau kombinasinya, dengan nilai PSNR di atas 21dB untuk standar deviasi 50. Semakin tinggi level noise, penggunaan filter rekursif atau kombinasi menghasilkan PSNR yang lebih baik.

Kata Kunci: filter mean, filter median, noise impulse, noise Gaussian

KATA PENGANTAR



Assalammu'alaikum wr wb.

Alhamdulillah rabbil'alam, penulis ucapkan sebagai tanda syukur yang sebesar-besarnya kepada Allah SWT, atas segala karunia dan rahmat yang diberikan-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik. Shalawat serta salam terucap buat junjungan Baginda Rasulullah Muhammad SAW, karena jasa Beliau kita bisa menikmati zaman yang penuh dengan ilmu pengetahuan seperti sekarang ini.

Tugas akhir ini disusun sebagai salah satu syarat untuk mendapatkan gelar kesarjanaan pada jurusan Teknik Informatika Universitas Islam Negeri Sultan Syarif Kasim Riau. Penulisan dan penyusunan laporan tugas akhir ini tidak terlepas dari bantuan dari berbagai pihak yang telah memberikan masukan-masukan kepada penulis. Pada kesempatan ini, penulis mengucapkan banyak terima kasih kepada :

1. Bapak Prof. DR. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Ibu Dra. Hj.Yenita Morena, M.Si, selaku Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Ibu Dr. Okfalisa, ST, M.Sc , selaku Ketua Jurusan Teknik Informatika.
4. Bapak Surya Agustian, ST, M.Kom, selaku Pembimbing tugas akhir yang telah meluangkan banyak waktunya untuk membimbing penulis dalam menyelesaikan Tugas Akhir ini.
5. Bapak Reski Mai Candra, ST, M.Sc, selaku Penguji I tugas akhir dan juga selaku Koordinator tugas akhir Jurusan Teknik Informatika yang telah banyak memberi masukan dan juga kemudahan dalam pengurusan Tugas Akhir.
6. Ibu Elvia Budianita, ST, M.Cs, selaku Penguji II tugas akhir dan juga selaku pembimbing akademik yang telah banyak memberi masukan dan juga kemudahan dalam pengurusan Tugas Akhir

7. Orang tuaku tercinta yang selalu memberikan doa, motivasi, bimbingan yang tiada hentinya, serta telah banyak berkorban demi keberhasilan anak-anaknya. Semoga mereka selalu dalam lindungan Allah SWT dan segala pengorbanan yang mereka berikan mendapat pahala dari Allah SWT. Aamiin.
8. Saudara-saudaraku, yang telah memberikan semangat dalam menyelesaikan tugas akhir ini.
9. Sahabat-sahabatku teman-teman seperjuangan Jurusan Teknik Informatika angkatan 2006 UIN Suska Riau. Semoga kita selalu diberi kelancaran oleh Allah dalam menggapai cita-cita dan menjadi insan yang berhasil. Aamiin.
10. Seluruh pihak yang belum penulis cantumkan, terima kasih atas dukungannya.

Penulis menyadari bahwa dalam penulisan laporan ini masih banyak kesalahan dan kekurangan. Oleh karena itu, kritik dan saran yang sifatnya membangun sangat diharapkan untuk kesempurnaan laporan ini. Akhirnya, penulis berharap semoga laporan ini dapat memberikan sesuatu yang bermanfaat bagi siapa saja yang membacanya. Amin.

Wassalamu'alaikum wr.wb.

Pekanbaru, 24 Juni 2013

M.HAKIKI OCEANDRA
10651004381

DAFTAR ISI

LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN.....	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK.....	vii
ABSTRACT.....	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvi
BAB I PENDAHULUAN	I-1
1.1 Latar Belakang	I-1
1.2 Rumusan Masalah	I-2
1.3 Batasan Masalah	I-2
1.4 Tujuan Penelitian	I-3
1.5 Sistematika Penulisan	I-3
BAB II LANDASAN TEORI.....	II-1
2.1 Citra Digital	II-1
2.2 Noise Pada Citra	II-3
2.2.1 Model Noise	II-5
2.2.2 Impulse Noise	II-6
2.2.3 Gaussian Noise.....	II-7
2.3 Rekonstruksi dan Restorasi Citra.....	II-8
2.3.1 Filter Spasial	II-8
2.3.2 Filter Spasial dengan Metode Mean	II-11
2.3.3 Filter Spasial dengan Metode Median	II-11
2.3.4 Filter Rekursif	II-12

2.3.5	Filter Kombinasi	II-12
2.4	Metode Pengukuran	II-13
2.5	Simulasi Perhitungan Manual	II-14
2.5.1	Simulasi Mean Filter	II-15
2.5.2	Simulasi Median Filter	II-17
BAB III METODOLOGI PENELITIAN		III-1
3.1	Perumusan Masalah	III-2
3.2	Studi Literatur	III-2
3.3	Analisa dan Perancangan	III-2
3.4	Implementasi dan Pengujian	III-3
3.5	Analisa	III-3
3.6	Simpulan	III-4
BAB IV ANALISA DAN PERANCANGAN		IV-1
4.1	Analisa Model Sistem	IV-1
4.1.1	Model Degradasi	IV-2
4.1.2	Model Restorasi	IV-2
4.1.3	Model Pengukuran Kualitas Filter	IV-3
4.2	Perancangan Sistem Filter	IV-5
4.2.1	Program Utama	IV-5
4.2.2	Proses Input Parameter	IV-7
4.2.3	Proses Menampilkan Citra Asli	IV-10
4.2.4	Proses Mengenakan dan Menampilkan Citra Bernoise	IV-10
4.2.5	Proses Eksekusi Filter dan Menampilkan Output	IV-10
BAB V IMPLEMENTASI DAN PENGUJIAN		V-1
5.1	Implementasi	V-1
5.1.1	Lingkungan Implementasi	V-1
5.1.2	Skema Implementasi	V-1
5.2	Pengujian dan Hasil Analisa Pengujian	V-2
5.2.1	Data Pengujian	V-3
5.2.2	Analisa Hasil Pengujian	V-11

5.2.3	Kesimpulan Hasil Pengujian.....	V-17
BAB VI PENUTUP.....		VI-1
6.1	Kesimpulan	VI-1
6.2	Saran	VI-2
DAFTAR PUSTAKA		
LAMPIRAN		
DAFTAR RIWAYAT HIDUP		

BAB I

PENDAHULUAN

1.1. Latar Belakang

Noise (derau) pada citra digital dapat terjadi karena banyak faktor, seperti kurangnya pencahayaan saat pengambilan gambar, keterbatasan resolusi *pixel* dari kamera yang digunakan, keterbatasan kemampuan menangkap gambar bergerak oleh kamera pengawas atau cctv (*closed circuit television*) yang disebabkan oleh terbatasnya memori dan buffer, interferensi gelombang elektromagnetik pada peralatan pencitraan kedokteran, dan sebagainya.

Noise mengganggu karena mengurangi kualitas citra pada saat pencetakan, menyulitkan identifikasi pelaku kejahatan pada citra cctv, juga menyulitkan pada deteksi penyakit atau sel kanker pada citra kedokteran (MRI, CTScan, XRay). Untuk itu diperlukan suatu metode pengurangan noise pada citra tersebut. Dalam beberapa aplikasi seperti pemrosesan citra kedokteran maupun penginderaan jauh melalui citra satelit, pengurangan noise menjadi syarat penting yang harus dilakukan sebelum tahap pemrosesan selanjutnya. Fase tersebut merupakan tahap preprocessing yang harus dilakukan untuk meningkatkan kualitas citra (*image enhancement*).

Banyak penelitian tentang metode pengurangan noise sampai saat ini, dan masih akan terus berlanjut sebagai pengembangan teknik pemrosesan citra digital, seperti yang dilakukan oleh James Church (2008), yang mengembangkan *spatial median filter* untuk mengurangi noise pada citra digital. Murinto dkk (2007) melakukan analisis perbandingan metode *intensity filtering* dengan *frequency filtering* pada aplikasi pengurangan noise pada citra digital. Pawan Patidar dkk (2010) mencoba beberapa variasi filter untuk mengurangi noise yaitu *Wavelet*

Transform filter, Mean Filter, Median Filter dan Wiener Filter. Tuti Adi Prihatin (2010) melakukan analisis penggunaan *low pass filter* untuk mengurangi noise pada citra digital dalam tugas akhirnya. Gouping Qiu (1996) mengembangkan filter rekursif berbasis median filter untuk tujuan yang sama. Masing-masing metode memiliki kelebihan dan kekurangan, karena pada akhirnya metode tertentu hanya cocok diaplikasikan untuk tipe noise tertentu pula, dan tingkat kepuasan terhadap hasil perbaikan secara kualitatif kembali kepada pengguna yang subjektif.

Tugas akhir ini mencoba menguji dan menganalisa dan mengombinasikan median filter, mean filter dan filter rekursif untuk mengatasi beberapa tipe noise pada citra digital. Semua filter yang diujicoba akan dibandingkan performanya melalui pengamatan kualitatif dan menghitung secara kuantitatif nilai PSNR (Peak Signal to Noise Ratio)-nya.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, dapat dipersempit pokok permasalahan dalam hal tugas akhir ini adalah “Bagaimana menghilangkan noise dengan menggunakan Mean Filter, Median Filter, kombinasi dan filter rekursif”.

1.3 Batasan Masalah

Batasan-batasan masalah dalam penulisan Tugas Akhir ini, agar fokus kepada tujuan penelitian adalah :

1. Simulasi dilakukan dengan menggunakan MATLAB
2. Tipe noise yang diberikan kepada citra asli adalah noise *impulse noise* (salt & pepper), dan *additive noise* (Gaussian).
3. Setiap tipe noise, dilakukan pengujian terhadap beberapa level intensitas noise
4. Ukuran performa filter secara kuantitatif adalah PSNR.

5. Pengamatan performa secara kualitatif dengan membandingkan citra output filter secara kasat mata (pengamatan).
6. Citra uji adalah citra standard berwarna yang dipakai pada riset perbaikan kualitas citra di dunia (*lena* dan *baboon*), dan citra dari koleksi pribadi (pas foto).

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai penulis dari Tugas Akhir ini adalah untuk menganalisa performa Mean Filter, Median Filter, kombinasi dan rekursif filter untuk menghilangkan impulse noise dan additive noise, yang dilakukan secara kualitatif dengan membandingkan citra asli, citra terkena noise dan citra output dari filter, dan secara kuantitatif dengan membandingkan nilai PSNR-nya. Efek noise diberikan terhadap gambar asli yang bersih (clean image) dengan menggunakan fungsi noise yang tersedia pada Matlab.

1.5 Sistematika Penulisan

Tugas akhir ini disusun dengan sistematika sebagai berikut :

Bab I Pendahuluan

Bab ini menerangkan latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan sistematika penulisan dari Tugas Akhir yang dibuat.

Bab II Landasan Teori

Dasar teori-teori mengenai citra dan filter yang berhubungan dengan tugas akhir ini diterangkan pada Bab II.

Bab III Metodologi Penelitian

Langkah-langkah yang dilaksanakan dalam Tugas Akhir ini, dimulai dari identifikasi masalah, perumusan masalah, mengumpulkan referensi, melakukan simulasi awal (studi literature), membuat program simulasi filter dengan menggunakan matlab, melakukan pengujian, ditutup dengan kesimpulan dan saran.

Bab IV Analisa dan Perancangan

Tahapan ini berisi analisa dan perancangan program filter yang akan dibuat dengan menggunakan matlab.

Bab V Implementasi dan Pengujian

Bab ini berisi pengujian performa filter dengan menggunakan beberapa citra standar dalam penelitian image processing, dan membandingkannya untuk melihat mana yang terbaik. Analisa performa secara kualitatif melalui pengamatan langsung dan kuantitatif melalui nilai PSNR.

Bab VI Kesimpulan dan Saran

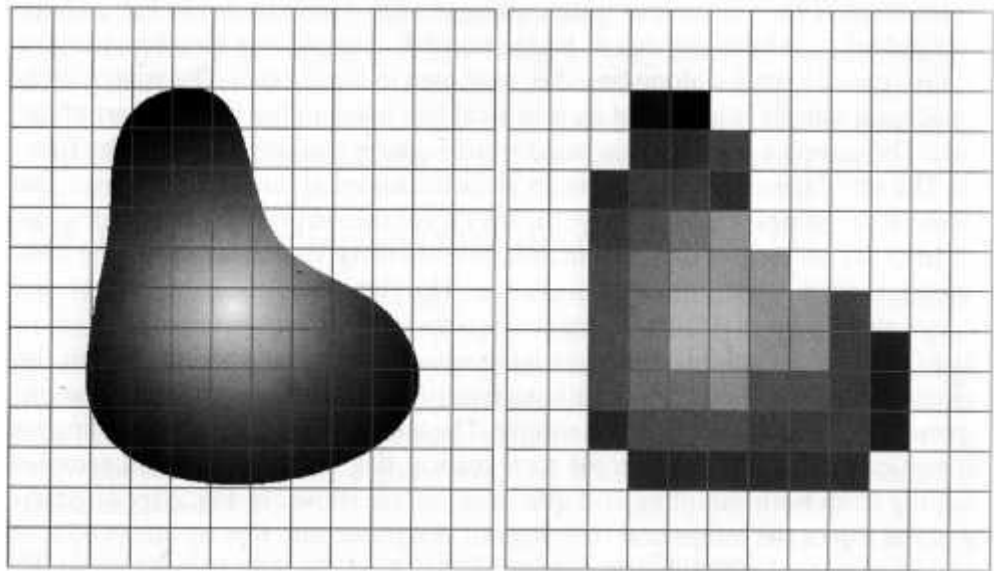
Bagian ini berisi kesimpulan dan saran mengenai penelitian ini.

BAB II

LANDASAN TEORI

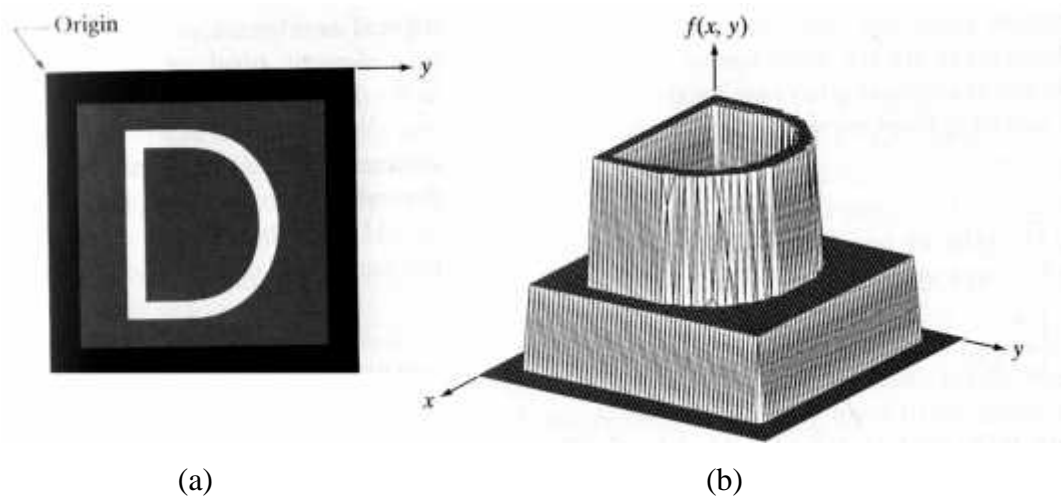
2.1 Citra Digital

Suatu citra (gambar) analog dengan ukuran panjang kali lebar, dapat didigitalisasi dengan mengambil sampling berupa matriks berukuran m kali n , dengan m adalah jumlah sampling untuk panjang, dan n adalah jumlah sampling untuk lebar. Setiap sampling adalah berukuran bujur sangkar kecil. Semakin kecil ukuran sampling tersebut, semakin baik representasi citra kedalam bentuk digital, dan semakin halus tepian (*edge*) gambar yang dihasilkan. Gambar 2.1 dibawah ini menunjukkan contoh suatu gambar yang didigitalisasi dengan ukuran sampling yang masih cukup besar, sehingga tepian gambar akan berbentuk kasar (kotak-kotak).



Gambar 2.1 Citra kontinu (analog) disampling oleh array sensor kuantisasi
(Sumber, Gonzales, 2008)

Sampling tersebut (selanjutnya disebut piksel, atau dalam bahasa Inggris adalah picture elemen = pictel menjadi pixel) yang terkecil dinyatakan dengan dot (atau berbentuk titik) karena ukuran yang sangat kecil. Matriks dot yang dihasilkan menyatakan derajat keabuan dari nilai sampling tersebut, untuk citra 8 bit bernilai 0 sampai 255. Untuk lebih jelas, citra digital seperti gambar 2.2 yang berbentuk suatu derajat keabuan (*gray scale*) dapat diplot pada grafik untuk memperlihatkan di level mana derajat keabuannya berada pada setiap piksel.



Gambar 2.2 Citra asli (a) Citra diplot pada suatu permukaan (b),

Sumber (Gonzalez, 2008)

$F(x,y)$ merupakan representasi dari nilai piksel pada setiap titik koordinat (x,y) , dengan titik *origin* dari matriks citra adalah $(0, 0)$ di mana x dan y masing-masing adalah baris dan kolom.

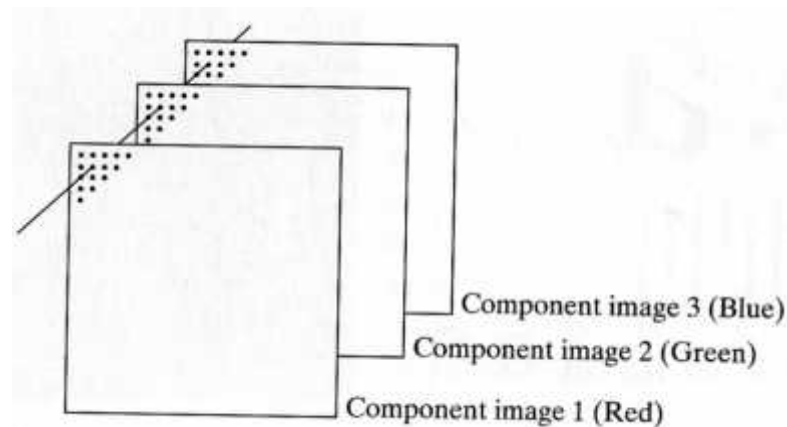
Untuk memudahkan penulisan matematis, maka matriks citra tersebut dapat dituliskan dalam bentuk persamaan (2.1) berikut ini.

$$F(x,y) = A_{x,y} = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \dots & a_{M-1,N-1} \end{bmatrix} \quad (2.1)$$

Bila citra adalah citra berwarna, maka warna direpresentasikan oleh kanal kanal RGB (red-green-blue) seperti gambar 2.3, dengan setiap kanal dinyatakan dengan derajat keabuan dengan rentang 0 - 255. Misalnya untuk red, 0

menyatakan paling merah (pekat) dan 255 menyatakan paling pudar (putih). Demikian juga untuk kanal green (hijau) dan blue (biru).

Sehingga matriks citra akan berdimensi ($m \times n \times o$) dengan m dan n adalah dimensi panjang dan lebar atau dimensi baris dan kolom pada matriks, dan o adalah dimensi untuk kanal warna, yaitu 1 untuk kanal RED (merah), 2 untuk GREEN (hijau) dan 3 untuk BLUE (biru).



Gambar 2.3 Ilustrasi representasi kanal warna RGB pada citra

2.2 Noise pada Citra

Noise (derau) merupakan penyebab utama penurunan kualitas citra (degradasi), sehingga sangat mengganggu apabila suatu foto ingin dicetak dan disimpan pada album atau dipajang diruang tamu. Dalam beberapa aplikasi medis, citra hasil pemindai MRI, CT Scan maupun USG, juga dapat terkena noise. Terlebih lagi USG, citra hitam putih analog yang dihasilkan sangat banyak memiliki noise, sehingga bila dibutuhkan analisa berbantuan komputer, perlu dilakukan preproses untuk memperbaiki kualitas citra (*enhancement*), agar deteksi selanjutnya bisa lebih tepat atau presisi.

Ada 3 tipe noise yang umum pada pemrosesan citra digital, yaitu impulse noise, additive noise dan multiplicative noise. Ketiga tipe ini dapat dijelaskan sebagai berikut :

- a. Impulse noise, merupakan noise yang berbentuk sinyal impuls acak dan terdistribusi secara acak pula pada suatu citra digital. Adanya sinyal

impuls ini menyebabkan diskontinuitas pada suatu segmen citra, atau pada suatu spatial window yang dievaluasi.

Contoh impulse noise adalah salt and pepper seperti diilustrasikan pada Gambar 2.4 (a).

- b. Additive noise, adalah sinyal-sinyal dengan magnitude acak yang terdistribusi secara Gauss pada suatu citra digital.

Contoh additive noise adalah derau putih (*white noise*) dan Gaussian noise, seperti diilustrasikan pada Gambar 2.4 (b).

- c. Multiplicative noise, adalah suatu multiplikasi atau konvolusi dari beberapa noise dengan magnitude, distribusi dan intensitas yang berbeda.

Contoh speckle noise seperti ditunjukkan pada Gambar 2.4 (c).



(a) Impulse Noise



(b) Additive Noise



(c) Multiplicative Noise



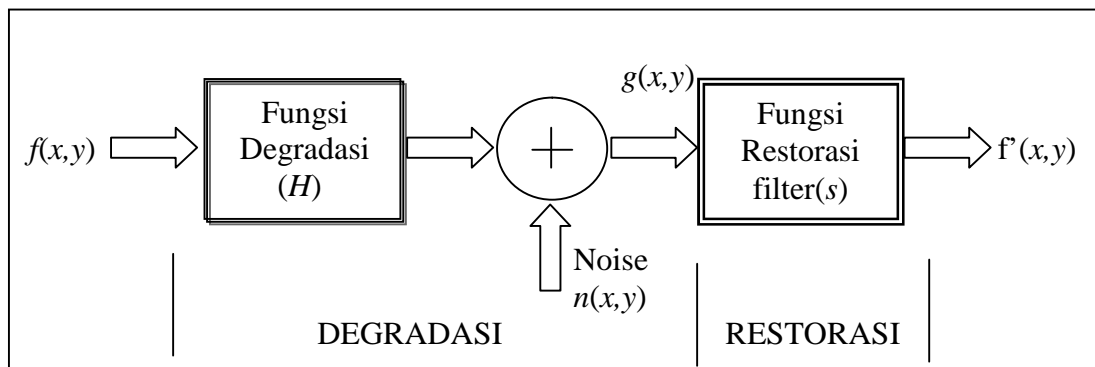
(d) citra asli

Gambar 2.4. Beberapa jenis noise menurunkan kualitas citra digital
(sumber: cover film Habibie & Ainun, 2013)

2.2.1 Model Noise

Menurut Gonzales (2008), sumber dasar dari noise dalam citra digital muncul selama pengambilan citra (image acquisition) kemudian mendigitalisasikannya (digitization) atau mengirimkan (transmission). Performa dari sensor citra dipengaruhi oleh banyak faktor yang berbeda, seperti kondisi lingkungan (cerah, terang atau kurang cahaya), dan kualitas dari elemen sensor pencitraan sendiri. Sebagai contoh, memperoleh citra dengan suatu kamera CCD, level pencahayaan, dan sensor temperature adalah faktor utama yang menentukan berapa besar noise yang dihasilkan pada citra yang dihasilkan. Citra yang terkorupsi selama transmisi pada dasarnya disebabkan karena adanya interferensi pada kanal yang digunakan untuk transmisi, misalnya transmisi menggunakan kanal tembaga secara analog, kemudian didigitalisasi, maka citra yang dihasilkan tentu akan terdegradasi.

Untuk memberikan efek noise pada suatu penelitian di bidang citra digital, maka suatu citra yang bersih dikenakan noise dengan memodifikasi setiap piksel di dalam citra melalui suatu operasi matematika. Dan kemudian, proses perbaikan citra dilakukan dengan merestorasi kembali setiap piksel dengan menggunakan filter tertentu, seperti terlihat pada Gambar 2.5 berikut ini.



Gambar 2.5. Model dari proses degradasi dan restorasi citra (Gonzales, 2008)

Dari gambar 2.5 di atas, bahwa citra asli yang bersih direpresentasikan dalam nilai piksel $g(x,y)$ untuk setiap posisi koordinat x dan y pada citra. Sedangkan model noise direpresentasikan oleh fungsi noise $n(x,y)$. Citra yang terdegradasi oleh noise, seperti terlihat pada persamaan (2.2) dihasilkan dari

operasi tertentu (disimbolkan dengan tanda “ \boxplus ”) yang direpresentasikan oleh fungsi (H) sehingga menghasilkan citra bernoise (*noisy image*) $g(x,y)$. Fungsi tersebut dapat berupa penjumlahan (*additive noise*), maupun mengubah nilai piksel (*replacement*) dengan fungsi noise yang ada.

$$g(x,y) = f(x,y) \boxplus n(x,y) \quad (2.2)$$

2.2.2 Impulse Noise

Untuk memodelkan Impulse noise dibangkitkan dengan menggunakan fungsi RAMP, yaitu suatu nilai diskontinu dari suatu piksel pada titik pusat di mana area sekelilingnya memiliki nilai piksel yang homogen.

Gonzales (2008) memodelkan Impulse Noise (Salt and Pepper) secara matematis, sebagai berikut:

Piksel citra dengan impulse noise bipolar (*black and white*) terdegradasi pada $P(z)$ dituliskan sebagai berikut:

$$P(z) = \begin{cases} P_a & \text{untuk } z = a \\ P_b & \text{untuk } z = b \\ 0 & \text{selainnya} \end{cases} \quad (2.3)$$

Bila $b > a$, intensitas dari b akan muncul sebagai titik terang cahaya pada citra. Kebalikannya, level a akan muncul sebagai titik gelap pada citra. Bila P_a atau P_b adalah nol, impulse noise disebut *unipolar*. Bila tidak mungkin probabilitasnya nol, khususnya bila keduanya kurang lebih seimbang, nilai *impulse noise* akan membentuk taburan garam dan merica (*salt and pepper*) secara *random* yang terdistribusi pada citra.

Noise yang berbentuk impulse dapat bernilai positif atau negative, artinya dia bisa berupa penambahan nilai piksel pada citra asli, maupun dapat mengurangi nilai citra asli. Penskalaan sebagai bagian dari proses pendigitaliasian citra digunakan, karena biasanya citra yang terkorupsi dapat lebih besar dari nilai piksel maksimum atau lebih kecil dari nilai piksel minimum (di luar rentang derajat keabuan 0-255). Impulse noise didigitalisasi dengan nilai yang ekstrim seperti sangat hitam (0 pada derajat keabuan) atau sangat putih (255 pada derajat keabuan).

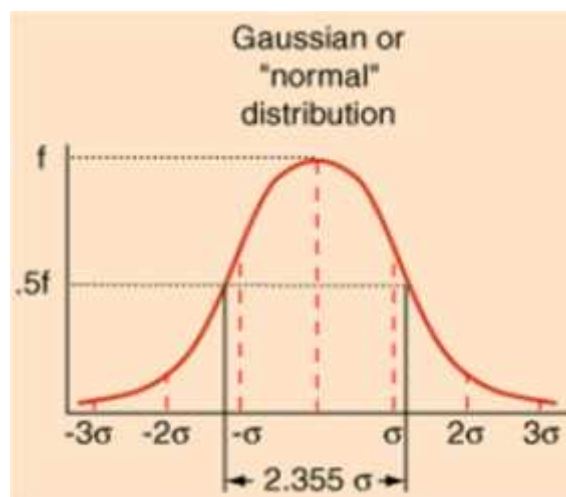
Hal ini diasumsikan bahwa nilai piksel yang terkena noise adalah tersaturasi pada warna hitam atau warna putih skala derajat keabuan tersebut. Oleh sebab itu, impulse noise disebut sebagai *salt and pepper*, di mana bila titik noise adalah putih, diumpamakan sebagai taburan garam (*salt*) dan bila berwarna hitam, merupakan taburan merica (*pepper*).

2.2.3 Gaussian Noise

Noise dengan distribusi Gauss (atau biasa juga disebut distribusi normal) dimodelkan dengan formula sebagai berikut (Gonzales 2008),

$$P(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2} \quad (2.4)$$

Dengan z adalah intensitas, dan \bar{z} adalah rata-ratanya, serta σ adalah standar deviasi dari distribusi Gauss. Kuadrat dari standard deviasi (σ^2) disebut sebagai varain dari z . Gambar 2.6 berikut ini mendeskripsikan contoh fungsi distribusi Gauss dengan parameter-parameter tersebut, dengan perkiraan bahwa 70% dari nilai pikselnya berada di antara rentang $[(\bar{z} - \sigma), (\bar{z} + \sigma)]$. Dan f adalah puncak grafik distribui Gauss yang dicapai pada nilai $z = \bar{z}$ (rata-rata), dengan besarnya adalah $f = \frac{1}{\sqrt{2\pi}\sigma}$



Gambar 2.6. Contoh grafik fungsi distribusi Gauss

(sumber [http:// hyperphysics.phy-astr.gsu.edu](http://hyperphysics.phy-astr.gsu.edu))

2.3 Rekonstruksi dan Restorasi Citra

Dalam peningkatan citra (*image enhancement*), tujuan utama dari teknik restorasi adalah untuk meningkatkan citra dalam arti yang telah ditetapkan, seperti mempertajam (*sharpening*), menilangkan derau (*noise removing*), pengaturan cahaya (*brightness*) dan lainnya. Meskipun ada bidang yang tumpang tindih, peningkatan citra sebagian besar merupakan proses yang subyektif, sedangkan restorasi citra adalah untuk sebagian besar proses obyektif. Restorasi mencoba untuk memulihkan gambar yang telah terdegradasi (menurun kualitasnya) dengan menggunakan pengetahuan atau metode-metode yang telah diketahui dari fenomena degradasi seperti bintik noise, blur (kabur) dan seterusnya (Gonzalez, 2008). Dengan demikian, teknik restorasi berorientasi ke arah gambar asli. Dalam hal ini menghilangkan *noise* (derau) adalah salah satu teknik restorasi citra yang masih banyak diteliti.

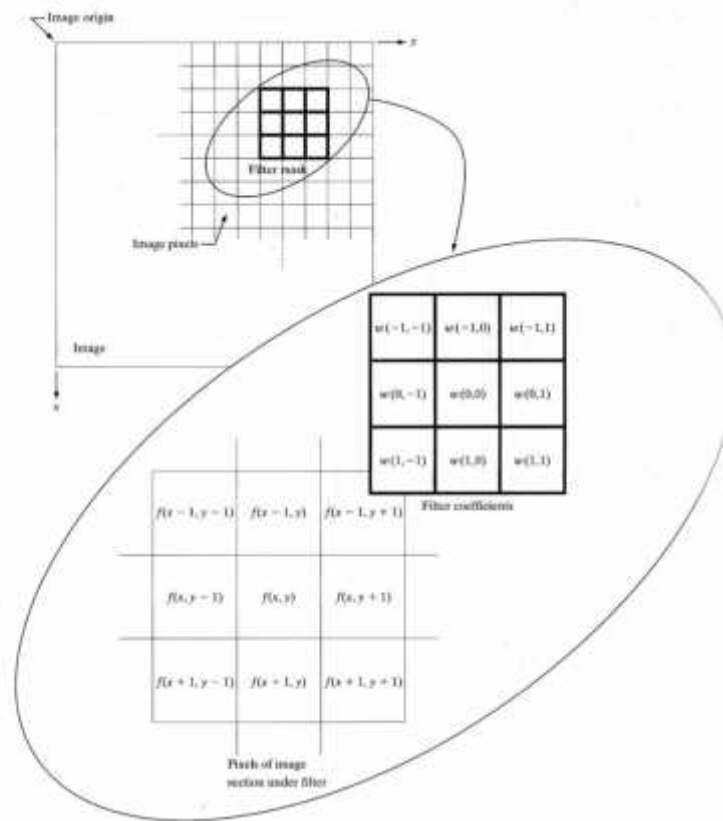
Pendekatan ini digunakan untuk merumuskan kriteria perbaikan yang akan menghasilkan perkiraan yang optimal untuk hasil yang diharapkan. Sebaliknya, teknik peningkatan pada dasarnya adalah prosedur heuristic (coba coba) yang dirancang untuk memanipulasi gambar untuk mendapatkan peningkatan dari gambar yang terdegradasi secara pengamatan langsung, sedangkan penghilangan efek gambar blur (kabur) dengan menerapkan fungsi deblurring dianggap sebagai **teknik restorasi**.

Salah satu pendekatan yang banyak digunakan, termasuk juga untuk teknik restorasi citra yang terkena noise adalah dengan menggunakan filter spasial. Filter ini bekerja dengan membentuk jendela spasial (*spatial window*) yang akan mengevaluasi setiap nilai piksel pada citra digital. Titik pusat piksel akan dikoreksi berdasarkan nilai piksel tetangganya. Lebih detil mengenai teknik ini dibahas pada sub bab berikut ini.

2.3.1 Filter Spasial

Mekanisme dari filter spasial adalah membentuk window bujur sangkar berukuran ganjil, seperti 3x3 piksel, 5x5, 7x7 dan seterusnya. Hal ini bertujuan untuk mendapatkan titik pusat piksel yang betul-betul berada di tengah-tengah

(centered). Titik pusat ini dievaluasi terhadap tetangganya, dan koreksi dilakukan misalnya berdasarkan kedekatan jarak antara piksel pusat dan piksel tetangga.



Gambar 2.7 Ilustrasi dari mekanisme filter spasial menggunakan window 3x3
(Sumber, Gonzales 2008)

Hal ini dapat diilustrasikan seperti Gambar 2.7 di atas. *Filter Mask* merupakan suatu himpunan nilai yang dihitung berdasarkan metode filtering yang digunakan. Beberapa metode statistic akan mengambil *filter mask* dari piksel-piksel yang dievaluasi, kemudian melakukan operasi statistic seperti menghitung mean (rata-rata), median (nilai tengah), atau modus (nilai yang sering muncul). Metode lainnya dapat juga berupa *weighted mask* yang dihitung dengan distribusi Gauss, *heuristic* dan lainnya.

Filter Mask menjadi dasar operasi konvolusi untuk proses koreksi dan restorasi citra, seperti yang telah digambarkan pada Gambar 2.7. Secara umum,

untuk ukuran window ganjil sembarang, misalnya $k=2n+1$, dengan $n=1,2,\dots$ dapat dituliskan formula untuk citra hasil koreksi sebagai mana persamaan (2.5) berikut,

$$F'(x,y) = \sum_{s=-n}^n \sum_{t=-n}^n w(s,t) g(x+s, y+t) \quad (2.5)$$

Di mana (x, y) adalah koordinat titik pusat piksel pada citra yang dievaluasi, w adalah *window mask*, dan g adalah piksel citra yang terdegradasi oleh noise.

Dalam perumusan, tidak ada salahnya untuk membentuk matriks *window filter* menjadi array 1 dimensi, demikian juga dengan sumber citra terdegradasi $g(x,y)$, sehingga memudahkan dalam proses evaluasi selanjutnya dalam metode statistic yang akan dibahas pada penelitian ini, sebagai berikut:

$$\begin{aligned} W_1 &= w(-1,-1) & G_1 &= g(x-1, y-1) \\ W_2 &= w(-1,0) & G_2 &= g(x-1, y) \\ W_3 &= w(-1,1) & G_3 &= g(x-1, y+1) \\ W_4 &= w(0,-1) & G_4 &= g(x, y-1) \\ W_5 &= w(0,0) & G_5 &= g(x, y) \\ W_6 &= w(0,1) & G_6 &= g(x, y+1) \\ W_7 &= w(1,-1) & G_7 &= g(x+1, y-1) \\ W_8 &= w(1,0) & G_8 &= g(x+1, y) \\ W_9 &= w(1,1) & G_9 &= g(x+1, y+1) \end{aligned} \quad (2.6)$$

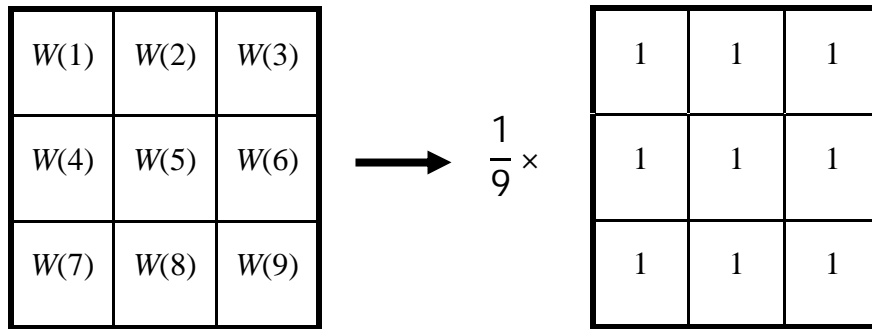
Selanjutnya, proses koreksi pada blok restorasi akan menghitung setiap piksel pusat window, yang dilakukan dengan menggunakan persamaan (2.6) yang merupakan penjabaran dari persamaan (2.5), seperti berikut ini,

$$\begin{aligned} F'(x,y) &= \sum_{r=1}^k W_r G_r \\ &= W_1 G_1 + W_2 G_2 + \dots + W_9 G_9 \end{aligned} \quad (2.7)$$

2.3.2 Filter Spasial dengan Metode Mean

Mean di dalam statistic adalah rata-rata. Dalam pengolahan citra, filter mean memiliki maksud menghitung nilai rata-rata dari suatu spasial window yang dievaluasi, kemudian nilai rata-rata tersebut menggantikan nilai piksel terdegradasi di dalam setiap window yang bergerak mulai dari origin sampai akhir. Secara umum, formula yang digunakan untuk output filter (F') adalah sebagai berikut:

$$F'_{x,y} = \frac{1}{k^2} \sum_{r=1}^k \sum_{r=1}^k W(r) G(r) \quad (2.8)$$



Gambar 2.8 window mask 3x3 untuk mean filter

Sebagai contoh, *Window Mask* untuk filter mean berukuran $k=3$, atau 3×3 , adalah matriks satuan, seperti terlihat pada Gambar 2.8, maka persamaan piksel pusat window hasil output dari filter adalah,

$$F'_{x,y} = \frac{G_1 + G_2 + \dots + G_9}{9} \quad (2.9)$$

2.3.3 Filter Spasial dengan Metode Median

Median adalah nilai tengah dari sekelompok data yang telah diurutkan. Dalam pengurangan noise pada citra, data-data yang diurutkan adalah nilai piksel di dalam spasial window, yaitu $W(1)$ sampai $W(k^2)$. Nilai tengah dari W yang terurut kemudian menjadi nilai baru bagi piksel pada titik pusat sebagai hasil

koreksi. Dalam hal ini, fungsi MEDIAN pada berbagai perangkat lunak pemrograman bisa dipakai, misalnya MEDIAN(W).

Namun demikian, algoritma menemukan nilai median dapat dituliskan sebagai berikut,

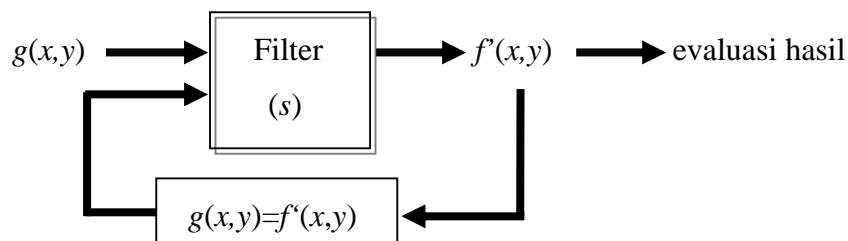
Langkah 1 : urutkan array window $W(r)$, dengan fungsi $W' = \text{sort}(W)$ dimana $r=1$ sampai k^2

Langkah 2 : cari indeks pertengahan r_t dengan rumus $r_t = \frac{1}{2}(k^2 + 1)$

Langkah 3 : Tukar nilai W pada pusat piksel dengan W' pada posisi r_t ,
 $W(r_t) = W'(r_t)$

2.3.4 Filter Rekursif

Rekursif bermakna pengulangan. Filter rekursif adalah filter yang menggunakan salah satu metode, kemudian outputnya difilter kembali dengan metode yang sama, seperti diilustrasikan pada Gambar 2.9 di bawah ini. Kita dapat membuat proses rekursif beberapa kali sesuai dengan kebutuhan dan tingkat kepuasan kita terhadap hasil yang diperoleh.



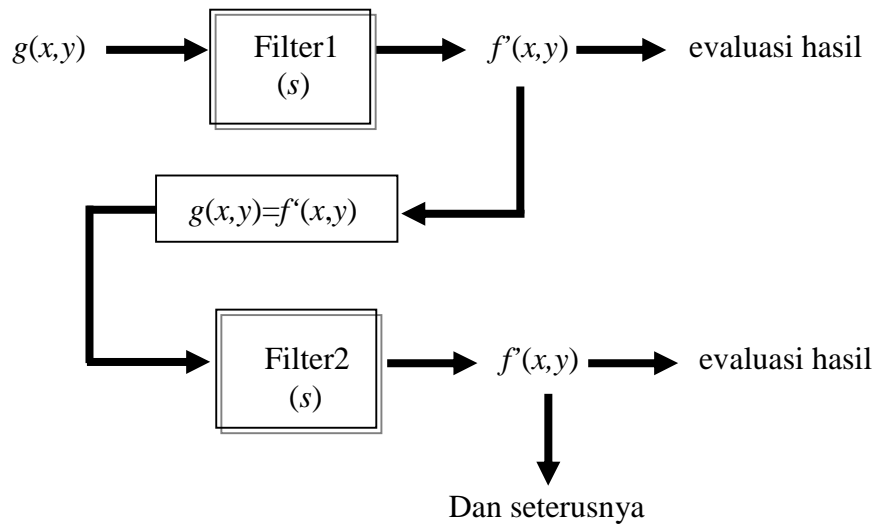
Gambar 2.9. Skema blok diagram sistem filter rekursif

2.3.5 Filter Kombinasi

Untuk mendapatkan hasil yang memuaskan, baik secara kualitatif terhadap pengamatan langsung citra output dari filter, maupun secara kuantitatif dengan mengukur nilai PSNR-nya masing-masing, beberapa metode filter dapat digabungkan dengan kombinasi tertentu.

Gabungan tersebut dapat berupa 2, 3, 4 filter berbeda dan seterusnya, dan yang mana yang dahulu pun dapat ditentukan secara coba-coba (heuristic). Secara

ringkas, proses filter kombinasi dapat digambarkan sebagai mana Gambar 2.10 berikut.



Gambar 2.10. Ilustrasi sistem kerja Filter Kombinasi

2.4 Metode Pengukuran

Beberapa citra standard yang digunakan sebagai *benchmark* (tolok ukur perbandingan) pada penelitian pengolahan citra berwarna digital, adalah Baboon (mewakili citra kaya warna dan detil) dan Lena (mewakili citra dengan warna sepia) seperti Gambar 2.11 (a) dan (c).

Untuk memperlihatkan kualitas perbaikan citra dari filter secara kualitatif, citra yang ditampilkan adalah bagian yang diperbesar (*zoom*) dari gambar asal, seperti ditunjukkan pada Gambar 2.11 (b) dan (d).



Gambar 2.11. *Benchmark Image* yang digunakan

Hasil objektif dari performa, dianalisa secara kuantitatif melalui nilai PSNR (Peak Signal to Noise Ratio), sebagai ukuran standard yang digunakan dalam pemrosesan citra. PSNR dihitung dari nilai MSE (*Mean Square Error*), yaitu nilai error rata-rata yang dihitung berdasarkan persamaan (2.9),

$$MSE(A, F') = \frac{1}{3MN} \sum_{s=1}^M \sum_{t=1}^N \sum_{c=1}^3 (A(s, t, c) - F'(s, t, c))^2 \quad (2.9)$$

dengan A adalah Citra Asli atau *clean image* (sebelum dikenai noise), dan F' adalah *filtered image* (citra yang terfilter atau citra yang terkena noise untuk yang belum difilter). Parameter s, t, c adalah indeks dimensi citra, yaitu berturut-turut ukuran baris (M), ukuran kolom (N) dan *color channel* (kanal warna, R=1, G=2, B=3).

PSNR memiliki satuan decibel (dB), diperoleh melalui persamaan (2.10) berikut,

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE(A, F')} \right) \quad (2.10)$$

Kualitas citra yang lebih baik adalah bila nilai MSE semakin kecil, yang berarti bahwa error yang terjadi atau selisih perbedaan setiap pixel pada setiap posisi (x, y) antara citra asli (*clean image*) dengan citra output dari filter semakin rendah. Dengan demikian, PSNR-nya akan semakin besar.

2.5 Simulasi perhitungan manual

Sepotong citra yang terkena oleh noise digambarkan dengan derajat keabuan sebagai berikut :

2	2	2	3	4	5	2	3
2	3	5	2	144	2	5	2
3	3	3	3	4	5	6	7
3	4	240	3	2	3	5	6
2	3	4	6	8	222	2	5
2	4	6	7	9	3	5	7

Didalam potongan gambar diatas ada tiga pixel yang warnanya berbeda dari sekitarnya yaitu pada indeks koordinat (2,5) sebesar 144 skala keabuan. Yang

kedua adalah pixel pada koordinat (4,3) sebesar 240 skala keabuan. Yang terakhir adalah pixel pada koordinat (5,6) sebesar 222 derajat keabuan.

Noise itu akan dihilangkan dengan filter mean dan median, dengan cara mengevaluasi setiap titik pada citra dengan membentuk window K x K dengan K adalah bilangan ganjil misalnya 3x3, 5x5 dan seterusnya. Titik yang dievaluasi menjadi pusat window.

2.5.1 Simulasi Mean Filter

Tujuannya untuk mencari rata-rata dari pixel-pixel pada window yang dievaluasi. Nilai ini selanjutnya menggantikan titik tengah window sebagai koreksinya. Window adalah area yang diarsir dengan titik tengah/pusat adalah piksel yang akan diperbaiki.

Dari contoh diatas, tahap 1 dibentuk window 3x3 dengan titik pusat di kordinat (1,1).

2	2	2	3	4	5	2	3
2	3	5	2	144	2	5	2
3	3	3	3	4	5	6	7
3	4	240	3	2	3	5	6
2	3	4	6	8	222	2	5
2	4	6	7	9	3	5	7

$$\begin{aligned}
 F_{1,1} &= \frac{2 + 2 + 2 + 2 + 3 + 5 + 3 + 3 + 3}{9} \\
 &= 2.78 \quad 3
 \end{aligned}$$

Maka nilai koreksi untuk pixel pada posisi (1,1) adalah 2,78 atau dibulatkan dalam derajat keabuan $F(1,1) = 3$.

Selanjutnya titik pusat window bergeser ke posisi (1,2). Langkah yang sama dilakukan untuk menghitung nilai rata-rata dari titik pusat window.

2	2	2	3	4	5	2	3
2	3	3	2	144	2	5	2
3	3	3	3	4	5	6	7
3	4	240	3	2	3	5	6
2	3	4	6	8	222	2	5
2	4	6	7	9	3	5	7

$$\begin{aligned}
 F_{1,2} &= \frac{2 + 2 + 3 + 3 + 5 + 2 + 3 + 3 + 3}{9} \\
 &= 2.86 \quad 3
 \end{aligned}$$

Maka nilai koreksi untuk pixel pada posisi (1,2) adalah 2,98 atau dibulatkan dalam derajat keabuan $F(1,2) = 3$.

Kemudian titik pusat window bergeser ke posisi (1,3). Langkah yang sama dilakukan untuk menghitung nilai rata-rata dari titik pusat window.

2	2	2	3	4	5	2	3
2	3	3	19	144	2	5	2
3	3	3	3	4	5	6	7
3	4	240	3	2	3	5	6
2	3	4	6	8	222	2	5
2	4	6	7	9	3	5	7

$$\begin{aligned}
 &F_{1,3} \\
 &= \frac{2 + 3 + 4 + 3 + 2 + 144 + 3 + 3 + 4}{9} \\
 &= 18.67 \quad 19
 \end{aligned}$$

Maka nilai koreksi untuk pixel pada posisi (1,3) adalah 18.67 atau dibulatkan dalam derajat keabuan $F(1,3)=19$.

Kemudian titik pusat window bergeser ke posisi (1,4). Langkah yang sama dilakukan untuk menghitung nilai rata-rata dari titik pusat window.

2	2	2	3	4	5	2	3
2	3	3	19	21	2	5	2
3	3	3	3	4	5	6	7
3	4	240	3	2	3	5	6
2	3	4	6	8	222	2	5
2	4	6	7	9	3	5	7

$$\begin{aligned}
 &F_{1,3} \\
 &= \frac{3 + 4 + 5 + 19 + 144 + 2 + 3 + 4 + 5}{9} \\
 &= 21
 \end{aligned}$$

Maka nilai koreksi untuk pixel pada posisi (1,4) adalah 21. Terlihat bahwa piksel yang tadinya berwarna mendekati abu-abu (pertengahan antara hitam dan putih, atau 0 dan 255), dikoreksi menjadi mendekati hitam, atau mendekati 0, yaitu bernilai 21.

Proses ini dilakukan terus, sampai posisi piksel terakhir, sehingga semua piksel pada citra berhasil dikoreksi.

Untuk citra berwarna yang terdiri dari 3 dimensi (kanal) warna, yaitu $m \times n \times o$, dengan o adalah indeks dimensi matriks untuk masing-masing kanal warna, yaitu RED=1, GREEN=2, dan BLUE=3. Masing-masing kanal merupakan citra dua dimensi $m \times n$ dengan derajat intensitas warna yang sama seperti proses pada grayscale (bernilai 0 sampai 255). Proses koreksi seperti di atas dilakukan untuk setiap kanal warna, dengan mengulangi evaluasi piksel pada pusat window, untuk setiap kanal R,G dan B.

2.5.2 Simulasi Median Filter

Tujuannya untuk mencari nilai tengah dari pixel-pixel pada window yang dievaluasi. Nilai ini selanjutnya menggantikan piksel pada titik tengah window sebagai koreksinya.

Dari contoh yang sama dengan mean filter diatas, tahap 1 dibentuk window 3x3 dengan titik pusat di kordinat (1,1).

2	2	2	3	4	5	2	3
2	3	5	2	144	2	5	2
3	3	3	3	4	5	6	7
3	4	240	3	2	3	5	6
2	3	4	6	8	222	2	5
2	4	6	7	9	3	5	7

$$F_{1,1} = \text{median}(2,2,2,2,3,3,3,3,5) \\ = 3$$

Median adalah nilai tengah
(indeks ke-5) setelah diurutkan
dari 9 data

Maka nilai koreksi untuk pixel pada posisi $F(1,1) = 3$, atau nilainya tidak berubah.

Selanjutnya titik pusat window bergeser ke posisi (1,2). Langkah yang sama dilakukan untuk menghitung nilai tengah dari titik pusat window.

2	2	2	3	4	5	2	3
2	3	3	2	144	2	5	2
3	3	3	3	4	5	6	7
3	4	240	3	2	3	5	6
2	3	4	6	8	222	2	5
2	4	6	7	9	3	5	7

$$F_{1,2} = \text{median } 2,2,2,3,3,3,3,3,5 \\ = 3$$

Maka nilai koreksi pixel pada posisi $F(1,2)$ adalah 3 dari nilai sebelumnya 5.

Kemudian titik pusat window bergeser ke posisi (1,3). Langkah yang sama dilakukan untuk menghitung nilai tengah dari titik pusat window.

2	2	2	3	4	5	2	3
2	3	3	3	144	2	5	2
3	3	3	3	4	5	6	7
3	4	240	3	2	3	5	6
2	3	4	6	8	222	2	5
2	4	6	7	9	3	5	7

$$F_{1,3} = \text{median}(2,2,3,3,3,3,4,4,144) \\ = 3$$

Maka nilai koreksi pixel pada posisi $F(1,3)$ adalah 3 dari sebelumnya bernilai 2.

Kemudian titik pusat window bergeser ke posisi (1,4). Langkah yang sama dilakukan untuk menghitung nilai tengah dari titik pusat window.

2	2	2	3	4	5	2	3
2	3	3	3	4	2	5	2
3	3	3	3	4	5	6	7
3	4	240	3	2	3	5	6
2	3	4	6	8	222	2	5
2	4	6	7	9	3	5	7

$$F_{1,3} = \text{median}(2,3,3,3,4,4,5,5,144) \\ = 4$$

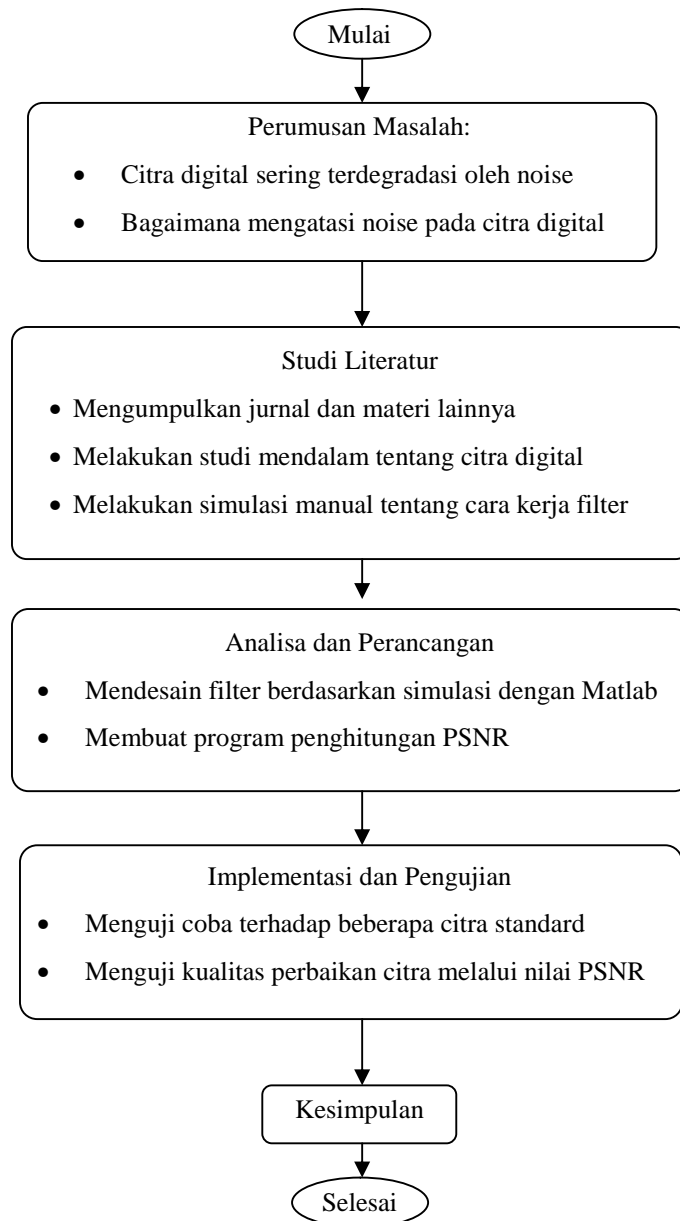
Maka nilai koreksi untuk pixel pada posisi $F(1,4)$ adalah 4. Terlihat bahwa piksel yang tadinya berwarna mendekati abu-abu (nilai 144) dikoreksi menjadi sangat mendekati hitam (0) yaitu bernilai 4.

Proses ini diulang sampai posisi piksel terakhir pada citra. Untuk citra berwarna, nilai piksel yang dikoreksi dilakukan untuk setiap kanal warna, atau indeks dimensi matriks R,G,B.

BAB III

METODOLOGI PENELITIAN

Penelitian mengenai perbaikan kualitas citra digital dengan menggunakan filter statistik ini, dilakukan dengan mengikuti langkah-langkah metodologi sebagai mana terlihat pada Gambar 3.1 berikut ini.



Gambar 3.1. Diagram alir penelitian

3.1 Perumusan Masalah

Pada tahap ini dilakukan analisa terhadap masalah, bahwa dalam citra digital, sering terjadi gangguan kualitas berupa noise. Masalah ini didalami dengan dengan membaca dan merujuk kepada berbagai sumber, bahwa noise yang mengganggu tersebut dapat dikurangi dan diminimalisir, dan sudah banyak kalangan mencoba meneliti permasalahan tersebut.

Dari pencarian referensi awal, maka dirumuskan masalah yang akan diangkat pada peneltian ini, yaitu untuk mengurangi noise pada citra digital dengan mendesain filter berbasis statistik, yaitu filter median, mean rekursif dan kombinasi.

3.2 Studi Literatur

Pada tahap ini dilakukan pengumpulan bahan-bahan dan referensi mengenai penelitian-penelitian sebelumnya, dan juga mempelajari buku teks mengenai pengolahan citra digital. Dalam mempelajari literatur, dilakukan tinjauan pustaka yang mendalam untuk memahami citra digital dan juga cara kerja filter spasial untuk mengoreksi citra yang terdegradasi.

Simulasi sederhana dilakukan untuk lebih memahami cara kerja filter dalam merekonstruksi nilai piksel pada citra digital yang terdegradasi oleh noise. Simulasi ini dilakukan secara manual dengan menggunakan Microsoft Excel untuk menghitung nilai-nilai piksel tersebut. Konsep dasar dan simulasi tersebut diuraikan pada Bab II.

3.3 Analisa dan Perancangan

Tahap perancangan dilakukan dengan membuat program filter dengan menggunakan Matlab. Algoritma program dibuat berdasarkan simulasi yang telah dilakukan sebelumnya dengan menggunakan Ms. Excel, menerapkan formula-formula dan menjalankan fungsi-fungsi matriks pada Matlab, untuk mengolah

nilai-nilai piksel citra digital sesuai dengan filter yang diterapkan. Lebih detilnya diuraikan pada Bab IV.

Kemudian dilakukan simulasi awal terhadap beberapa citra digital standard yang digunakan untuk pengolahan citra digital. Selanjutnya adalah melakukan analisis matematis untuk menghitung hasil perbaikan kualitas citra yang terdegradasi, dan dibandingkan dengan citra aslinya. Analisis tersebut dituliskan dalam rumus-rumus yang akan diimplementasikan pada Matlab.

3.4 Implementasi dan Pengujian

Implementasi program filter yang dilanjutkan dengan perhitungan kualitas hasil perbaikan melalui perhitungan PSNR, dilakukan terhadap beberapa citra uji standard dalam pengolahan citra. Selanjutnya dilakukan juga pengujian terhadap citra-citra populer lainnya, yang meliputi citra kaya warna (*rich color*) dan citra *sepia* (kesan *old picture*, warna dominan coklat dan merah).

Citra tersebut dikenakan noise berupa impulse (*salt and pepper*) dan *additive* (*Gaussian noise*) dengan berbagai level atau standard deviasi, dan dihitung PSNR-nya. Selanjutnya citra yang telah terkena noise ini dimasukkan ke dalam filter yang telah dibuat, dan data yang disimpan adalah hasilnya berupa data visual (citra hasil perbaikan) dan numerik (PSNR dari output filter).

3.5 Analisa

Hasil simulasi berupa data visual citra hasil perbaikan sebagai representasi citra dianalisa secara kualitatif, dan raw data (numerik) atau berupa matriks derajat keabuan dari masing-masing komponen warna RGB yang diperoleh akan dianalisa secara kuantitatif, yaitu dengan membandingkan nilai PSNR (Peak Signal-to-Noise Ratio) dari masing-masing filter terhadap citra asli.

Pada akhirnya, untuk melihat secara kualitatif bagaimana performa filter yang dibuat, maka semua diplot dalam satu grafik, sehingga kita dapat membandingkan performa masing-masing filter, yang mana yang terbaik terhadap

noise yang dikenakan. Filter terbaik tentu saja yang memiliki grafik tertinggi dibanding filter lainnya terhadap noise. Secara lengkap Analisa dilakukan pada Bab V.

3.6 Simpulan

Kesimpulan akhir dapat ditarik dari hasil eksperimen, dan diuraikan pada bagian akhir penelitian. Saran-saran untuk penelitian selanjutnya dapat diajukan dengan melihat hasil-hasil yang telah diperoleh pada penelitian ini.

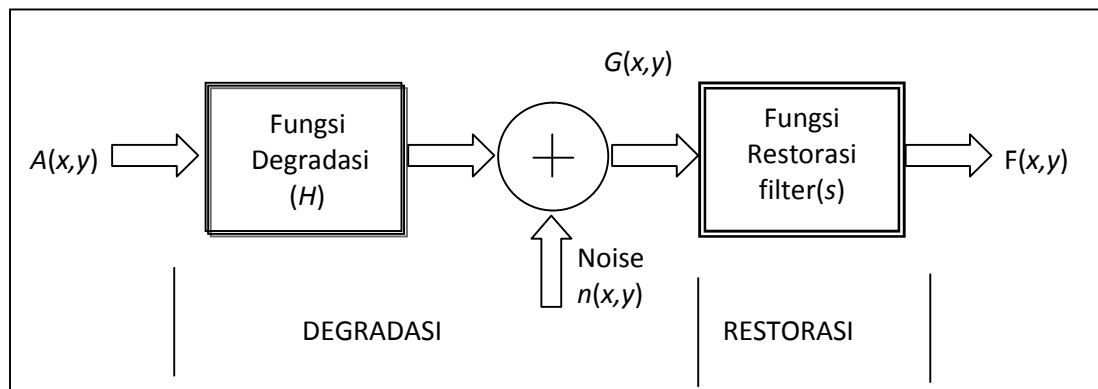
BAB IV

ANALISA DAN PERANCANGAN

Bab ini akan membahas analisa dan perancangan program noise filter dengan menggunakan Matlab. Analisa bertujuan untuk mengidentifikasi permasalahan, cara kerja dan output yang diharapkan dari sistem filter, sementara perancangan bertujuan untuk memodelkan masalah ke dalam program, kemudian membuat rutin program filter untuk memecahkan masalah noise tersebut, dan mempersiapkan aspek-aspek pengujiannya.

4.1 Analisa Model Sistem

Sistem pengurangan noise pada citra digital ini dibangun berdasarkan model penelitian standard pengolahan citra yang telah dikembangkan dan digunakan di seluruh dunia, dengan berpatokan pada Gambar 4.1 berikut ini (Gonzales, 2008).



Gambar 4.1. Model penelitian pengolahan citra

4.1.1 Model Degradasi

Dalam dunia nyata, citra yang diperoleh dari perangkat pencitraan seperti kamera dan sebagainya, sering terkena noise sehingga kualitas citra jadi menurun (terdegradasi). Sehingga, untuk memodelkan citra yang terdegradasi tersebut, maka dari citra asli yang bersih (*clean image*) $A(x,y)$, diberikan noise $n(x,y)$ menurut jenis noise tertentu (*impulse* atau *Gaussian*), dengan fungsi degradasi $H(s)$ adalah matriks satuan $H(1)$ dengan ukuran dimensi yang sama dengan ukuran dimensi citra asli, sehingga proses konvolusi antara $A(x,y)$ dengan $H(1)$ akan tetap menghasilkan $A(x,y)$, atau dengan kata lain, fungsi degradasi $H(s)$ tidak akan mengubah nilai piksel dari citra asli.

$$A(x,y) * H(1) = A(x,y)$$

Selanjutnya, dari model di atas, output dari fungsi degradasi ditambahkan dengan noise $n(x,y)$ sehingga menghasilkan citra yang terkorupsi oleh noise $G(x,y)$ sebagai berikut:

$$G(x,y) = A(x,y) + n(x,y)$$

Citra yang terdegradasi oleh noise atau $G(x,y)$ merupakan citra input bagi filter yang akan dibuat. Filter tersebut kemudian dapat diimplementasikan dalam berbagai perangkat lunak pengolahan citra, baik berupa program komputer maupun aplikasi yang menyatu dengan perangkat pencitraan seperti kamera digital dan lainnya.

4.1.2 Model Restorasi

Untuk memperbaiki kualitas citra yang terdegradasi oleh noise, seperti citra ber-noise hasil tangkapan kamera digital pada malam hari di ruang terbuka, digunakan filter noise dengan berbagai metode, antara lain seperti yang akan dibuat pada penelitian ini.

Input filter adalah citra ber-*noise* $G(x,y)$. Selanjutnya citra tersebut dengan metode filter tertentu akan diperbaiki piksel demi piksel seluruhnya, berdasarkan nilai piksel tetangganya. Dari Gambar 4.1 di atas, model fungsi restorasi adalah proses mengolah input $G(x,y)$ oleh fungsi $filter(s)$, dan menghasilkan output citra $F(x,y)$ yang relatif bersih dari noise.

$$F(x,y) = G(x,y) * filter(s)$$

Fungsi $filter(s)$ adalah fungsi yang akan dikembangkan dengan matlab, dan akan diberi nama sesuai dengan metode yang diusulkan dalam penelitian ini, yaitu:

- a. $filter_median(s)$ adalah nama fungsi untuk median filter
- b. $filter_mean(s)$ adalah nama fungsi untuk mean filter
- c. $filter_rekursif(s)$ adalah nama fungsi untuk rekursif filter
- d. $filter_kombinasi(s)$ adalah nama fungsi untuk kombinasi mean dan median

4.1.3 Model Pengukuran Kualitas Filter

Untuk menilai seberapa bagus hasil perbaikan citra, maka citra output filter akan dibandingkan dengan citra asli. Teknik pengukuran secara kuantitatif adalah mengukur seberapa banyak perbedaan nilai piksel demi piksel antara citra asli terhadap citra output dari filter tersebut. Semakin kecil perbedaan nilai pikselnya, maka semakin baik kinerja filter yang dihasilkan, dan semakin bersih citra output filter tersebut dari noise.

Nilai error piksel demi piksel untuk citra dengan ukuran $m \times n$ dapat dihitung sebagai berikut:

$$Error(x,y) = A(x,y) - F(x,y)$$

Untuk memudahkan pengukuran, maka dibutuhkan satu angka penilaian, bukan dalam bentuk matriks $Error(x,y)$. Oleh karena itu, nilai error seluruh piksel dijumlahkan, mulai dari piksel pada posisi origin $(x,y)=(1,1)$ sampai piksel terakhir $(x,y)=(m,n)$.

Karena adanya kecenderungan bahwa di satu posisi (x,y) nilai error bisa positif (citra output lebih tinggi derajat keabuannya daripada citra asli), dan di posisi lain nilai errornya bisa negatif, maka penjumlahan nilai error harus berupa nilai mutlak selisih, atau dalam bentuk square error (error kuadrat). Penggunaan square error lebih populer, sehingga kemudian dari seluruh piksel, dapat dihitung nilai error rata-rata setiap piksel (*mean square error*) antara F (citra output filter) terhadap A (citra asli), dengan rumus sebagai berikut:

$$MSE_{F',F} = \frac{1}{m \times n} \sum_{x=1}^m \sum_{y=1}^n A_{x,y} - F_{x,y}^2$$

Untuk citra berwarna yang terdiri atas 3 dimensi (Red, Green, Blue), maka MSE menggunakan rumus yang sama dengan penambahan dimensi matriks $z=3$, sebagai berikut:

$$MSE_{F',F} = \frac{1}{m \times n \times 3} \sum_{z=1}^3 \sum_{x=1}^m \sum_{y=1}^n A_{x,y} - F_{x,y}^2$$

Selain satuan ukuran MSE, satuan ukuran lainnya yang juga populer dan lebih banyak digunakan adalah PSNR (Peak Signal to Noise Ratio) dalam satuan logaritmik dB (decibel). PSNR adalah penurunan rumus MSE, yang mengukur berapa tingkat error dari skala puncak atau nilai terbesar derajat keabuan, yaitu 255. Sehingga pengukuran untuk PSNR adalah sebagai berikut:

$$PSNR = 10 \log \frac{255^2}{MSE}$$

Semakin besar nilai PSNR, berarti semakin kecil error yang terjadi, dan semakin baik kualitas output filter yang dihasilkan. Semakin rendah PSNR, maka semakin buruk output filter yang dihasilkan. Nilai PSNR yang terendah pada pengukuran yang diharapkan dalam pengujian ini adalah PSNR dari citra yang terkena noise terhadap citra asli.

4.2 Perancangan Sistem Filter

Program Matlab yang dibuat adalah berbasis teks, yang dapat dijalankan langsung pada console Matlab, yang terdiri atas rutin program utama dan fungsi-fungsi filter yang dipanggil oleh program utama saat menu filter dipilih. Output filter berupa citra yang sudah diperbaiki dari noise ditampilkan, berikut dengan nilai MSE dan PSNR dari masing-masingnya. Di samping itu, ditampilkan juga citra yang terkena noise beserta nilai PSNR dan MSE-nya.

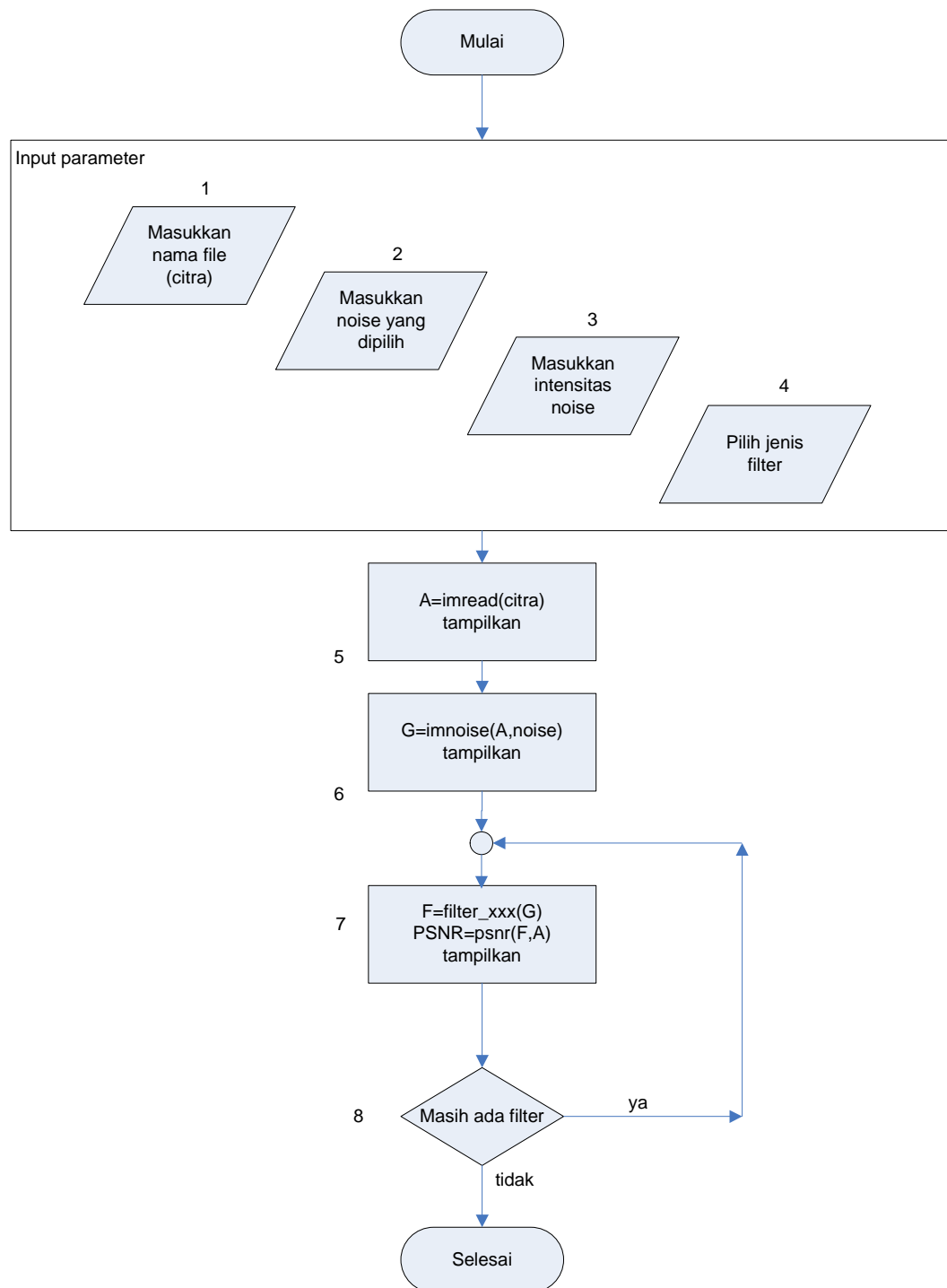
Pengamatan kualitatif dilakukan terhadap citra, dibandingkan secara berjejer dalam satu tampilan halaman agar dapat dilihat tingkat kualitasnya, seberapa baik output filter yang dihasilkan. Sedangkan pengukuran kuantitatif adalah dengan menganalisa nilai-nilai PSNR dalam suatu grafik untuk berbagai filter yang diterapkan.

4.2.1 Program Utama

Rancangan alur program utama sesuai diagram alir pada gambar 4.2 adalah sebagai berikut :

1. Masukkan nama file citra yang ingin diujikan
2. Pilih model noise yang akan dikenakan (impulse atau Gaussian)
3. Tentukan level noise yang dikenakan, seberapa besar intensitas noise memenuhi atau mengotori citra asli (secara default nilai tertentu diberikan).

4. Pilih filter yang akan diaplikasikan (secara default, seluruh filter dioperasikan)
5. Proses citra asli ke bentuk matriks, tampilkan citra asli



Gambar 4.2 Diagram alir program utama

6. Proses citra bernoise ke bentuk matriks, tampilkan citra bernoise
7. Proses filter yang dipilih, Hitung PSNR filter, Tampilkan citra output filter
8. Looping memproses seluruh filter yang dipilih.

4.2.2 Proses Input Parameter

File citra yang didukung oleh Matlab terdiri atas beberapa tipe, seperti jpeg, tiff, bmp, png dan lainnya, dapat dibaca melalui utilitas `imread`. Utilitas ini akan membaca nilai derajat keabuan masing-masing piksel penyusun citra ke dalam bentuk matriks. Untuk citra *grayscale*, dimensi matriks citra yang dihasilkan berdimensi $m \times n$. Sedangkan untuk citra berwarna, citra yang dibaca oleh utilitas `imread` menghasilkan dimensi berukuran $m \times n \times o$ dengan o adalah indeks untuk kanal warna RGB, (R=1, G=2, B=3).

Berikut adalah sintaks pada program utama yang digunakan untuk membaca file citra asli yang bersih dengan menggunakan utilitas `imread` (baris 3). Derajat keabuan file citra disimpan dalam variabel a (baris 3). Sedangkan ukuran file citra, disimpan dalam variabel m , n , dan o (baris 4).

```
1 X = input('\n Masukkan nama file gambar (filename.ext) : ');
2 ' , 's');
3 a = imread(X);
4 [m n o] = size (a);
```

Untuk pengenalan noise, user diminta untuk memilih noise yang akan dikenakan terhadap citra asli (baris 1-4). Ada beberapa tipe noise yang dapat digunakan dalam penelitian perbaikan kualitas citra, yaitu *impulse*, *additive* dan *multiplicative*. Dalam penelitian ini, hanya dibatas pada 2 macam tipe noise saja, yaitu *impulse* (Salt & Pepper) dan *additive* (Gaussian) noise (baris 2 dan baris 3). Apabila pengguna tidak memilih tipe noise, maka sistem akan menggunakan tipe noise Additive (Gaussian) (baris 5-6). Kode program yang dikembangkan untuk memilih noise di dalam Matlab adalah sebagai berikut.

```
1 fprintf('\n Pilih noise yang akan dikenakan : ')
2 fprintf('\n Ketik (1) untuk Gaussian (additive noise)')
3 fprintf('\n Ketik (2) untuk Salt & Pepper (impulse noise)')
```



```

4 noise = input('\n Nomor Pilihan Anda (default 1):');
5     if (isempty(noise))
6         noise=1;
7     end

```

Setelah jenis noise dipilih, maka user perlu untuk memasukkan ukuran atau intensitas noise (baris 13-15). Untuk noise dengan tipe impulse, maka intensitas noise adalah berapa persen noise yang dikenakan terhadap citra (satuan dalam persen dikonversi ke dalam desimal, misalnya noise 10% = 0.1).

Sedangkan noise dengan tipe Gaussian, parameter yang diinputkan adalah varians, dalam skala 0 – 255 (baris 3-5). Varians terbesar atau 255, maka tingkat noise yang dihasilkan akan maksimum. Biasanya ukuran noise tipe Gaussian yang masih mungkin untuk dikurangi memiliki nilai standar deviasi antara 0 sampai 50. Di atas itu, proses pengurangan noise sangat sulit, karena distribusinya yang merata. Nilai noise ini kemudian dinormalisasi dalam skala 100%, atau antara 0 dan 1 (baris 7). Karena fungsi Matlab untuk memberikan efek noise Gaussian harus menerima inputan berupa varians (kuadrat dari standar deviasi), maka varians dihitung seperti pada kode baris 8 dan baris 10. Berikut kode program untuk pemilihan jenis noise dan level atau intensitasnya.

```

1 switch (noise)
2     case 1
3         fprintf('\n Anda memilih Gaussian Noise. ');
4         V=input('\n Standard Deviasi (0-255, default =
5 30): ');
6         if (isempty(V))
7             V=30/255; %V di-normalisasi antara 0 sampai 1:
8             V=V^2;
9         else
10            V=(V/255)^2;
11        end
12    case 2
13        fprintf('\n Anda memilih Salt & Pepper Noise. ');
14        D=input('\n Density (isi dalam desimal, default 10% =
15 0.1): ');
16        if (isempty(D))
17            D=0.1;
18        end
19    otherwise
20        fprintf('\n Maaf, pilihan salah. Sistem memilih
21 Impulse noise intensitas 10% ');
22        noise=2; D=0.1;
23    end

```

Untuk memilih jenis filter yang akan digunakan dalam mereduksi noise, form inputan user disediakan sebagai mana kode berikut ini.

```
1 disp('Filter yang dapat digunakan dalam TA ini:');
2 disp('1. Filter median');
3 disp('2. Filter mean');
4 disp('3. Filter rekursif dari mean/median');
5 disp('4. Filter kombinasi dari mean/median');
6 filter = input('\n Pilih jenis filter (default 1):');
7     if (isempty(filter))
8         filter=1
9     end
10 switch (filter)
11     case 1
12         fprintf('\n Anda memilih Median Filter. ');
13     case 2
14         fprintf('\n Anda memilih Mean Filter. ');
15     case 3
16         fprintf('\n Anda memilih Filter Rekursif. ');
17         disp ('Basis filter untuk rekursif:');
18         disp ('1 = median');
19         disp ('2 = mean');
20         rekursif = input('\n Pilih basis filter untuk
21 rekursif:');
22         iterasi = input('\n Berapa kali iterasi? (jangan
23 banyak2 ya, nanti lama.):');
24
25     case 4
26         fprintf('\n Anda memilih Filter Kombinasi. ');
27         disp ('Basis filter untuk kombinasi:');
28         disp ('1 = median');
29         disp ('2 = mean');
30         disp ('Tulis kombinasi dalam bentuk matriks basis,
31 contoh:');
32         disp ('kombinasi median-mean = 12');
33         disp ('kombinasi mean-median = 21');
34         disp ('kombinasi median-mean-median = 121');
35         disp ('kombinasi mean-median-mean = 212');
36         kombinasi= input('\n Masukkan kombinasi filter
37 (jangan banyak2 yaaa.. ):');
38     otherwise
39         fprintf('\n Maaf,pilihan salah. Sistem memilih Filter
40 Median');
41         filter=1;
42 end
```

Dalam barisan kode di atas, hanya disimpan kode filter yang dipilih user, yaitu 1 sampai 4 untuk median, mean, rekursif dan kombinasi. Bila dipilih filter rekursif, maka disimpan lagi informasi basis filter (baris 20) dan jumlah iterasi (baris 22). Sedangkan bila dipilih kombinasi, maka informasi yang disimpan selanjutnya

adalah urutan basis filter yang dikombinasikan (baris 36). Selainnya adalah untuk menampilkan teks pada layar konsol Matlab.

4.2.3 Proses Menampilkan Citra Asli

Selanjutnya sistem akan menampilkan citra asli yang bersih dari noise, dengan perintah seperti kode berikut ini. Pada saat yang sama, informasi dimensi citra dan derajat keabuannya disimpan dalam variabel m , n , dan o (baris 2).

```
1 A = imread(X);
2 [m n o] = size (A);
3 figure;
4 imshow(A);
5 title('Gambar Asli');
```

4.2.4 Proses Mengenakan dan Menampilkan Citra Bernoise

Proses selanjutnya adalah mengenakan noise pada citra asli A, disimpan dalam variabel G (baris 2 dan baris 4), kemudian menampilkan citra yang terkena noise tersebut, sesuai dengan parameter noise yang telah diinputkan sebelumnya (baris 12). Proses tersebut dilakukan menurut kode berikut ini.

```
1 if (noise==1)
2     G = imnoise(A, 'gaussian', 0, V);
3 else if (noise==2)
4     G = imnoise(A, 'salt & pepper', D); %
5     end
6 end
7
8 fprintf('\n Noise:');
9 ppsn = psnr1(m,n,A,G);
10 Gnoise=uint8(G);
11 figure;
12 imshow(Gnoise); title('Gambar Terkena Noise');
```

4.2.5 Proses Eksekusi Filter dan Menampilkan Output

Proses utama dari program adalah mengolah citra yang terkena noise, kemudian mengurangi intensitas noise berdasarkan nilai-nilai piksel tetangganya.

Proses ini memanggil fungsi-fungsi filter sesuai dengan pilihan pengguna, yang diterangkan dalam bagian-bagian berikut ini.

4.2.5.1 Fungsi Filter Median

Fungsi filter Median dijalankan sebagaimana formula yang telah dituliskan pada Bab 2.3.3, yaitu langkah menghitung median dari piksel-piksel pada matriks window berukuran $k \times k$ yang telah ditransformasi menjadi matriks berukuran $1 \times k^2$ sesuai Persamaan (2.6) (baris 4-12). Langkah menghitung median pada program fungsi filter_median.m adalah sebagaimana baris 14-16 pada kode berikut ini.

```

1  for k = 1:z
2      for i = (1+1):(x-1)
3          for j = (1+1):(y-1)
4              W(1)=X((i-1),(j-1),k);
5              W(2)=X(i,(j-1),k);
6              W(3)=X((i+1),(j-1),k);
7              W(4)=X((i-1),j,k);
8              W(5)=X(i,j,k);
9              W(6)=X((i+1),j,k);
10             W(7)=X((i-1),(j+1),k);
11             W(8)=X(i,(j+1),k);
12             W(9)=X((i+1),(j+1),k); %persamaan 2.6
13
14             W=sort(W); %menghitung median pada bab 2.3.3
15             t=(9+1)/2
16             Xmed(i,j,k)=W(t);
17             end
18         end
19     end

```

4.2.5.2 Fungsi Filter Mean

Fungsi filter Mean menggunakan Persamaan 2.9 pada Bab 2.3.2, dari matriks window piksel berukuran 3x3 yang ditransformasi ke bentuk yang sama seperti sebelumnya (baris 4-12), yaitu sesuai Persamaan 2.6. Proses menghitung mean dari matriks window piksel tersebut sebagaimana baris 14-15) pada kode berikut ini.

```

1  for k = 1:z
2      for i = (1+1):(x-1)
3          for j = (1+1):(y-1)

```

```

4         W(1)=X((i-1),(j-1),k);
5         W(2)=X(i,(j-1),k);
6         W(3)=X((i+1),(j-1),k);
7         W(4)=X((i-1),j,k);
8         W(5)=X(i,j,k);
9         W(6)=X((i+1),j,k);
10        W(7)=X((i-1),(j+1),k);
11        W(8)=X(i,(j+1),k);
12        W(9)=X((i+1),(j+1),k); %persamaan 2.6
13
14        W=sum(W); %menghitung mean persamaan 2.9
15        Xmean(i,j,k)=W/9;
16        end
17    end
18 end

```

4.2.5.3 Filter Rekursif

Fungsi filter rekursif adalah menjalankan secara berulang filter median atau mean sebagai basis filter. Untuk keperluan tersebut, output dari setiap proses iterasi filter menjadi input bagi iterasi berikutnya. Oleh karena itu, untuk iterasi pertama, citra bernoise G disimpan dalam variabel F (baris 2), dan selanjutnya fungsi filter di dalam iterasi cukup mengambil variabel input F, dan disimpan dalam variabel output F pula (baris 6 dan 13), sebagaimana cuplikan program utama pada kode program di bawah ini.

Untuk menampilkan judul, maka string judul diproses pada variabel filt, yang menggabungkan beberapa string dalam bentuk array (baris 1, 5 dan 12). Basis filter yang dipilih pengguna disimpan dalam variabel bernama rekursif. Basis filter menentukan string filt yang akan ditampilkan (baris 5 dan 12) dan fungsi filter yang akan dipanggil (baris 6 dan 13), dalam hal ini:

Jika (rekursif=1), maka $F = \text{filter_median}(F)$

Selainnya $F = \text{filter_mean}(F)$

```

1         filt='';
2         F=G; %memindahkan variable gambar bernoise ke F
3         for i=1:iterasi
4             if (rekursif==1)
5                 filt=[filt ' Median'];
6                 F=filter_median(F);
7                 figure();imshow(F); title('Output Filter');
8                 fprintf(filt);

```

```

9         fprintf(' iterasi ke-');
10        fprintf('%d',i);
11    else
12        filt=[filt ' Mean'];
13        F=filter_mean(F);
14        figure();imshow(F); title('Output Filter');
15        fprintf(filt);
16        fprintf(' iterasi ke-');
17        fprintf('%d',i);
18    end
19    fprintf('\n output filter Rekursif ');
20    psnr=psnr1(A,F);
21    end

```

4.2.5.4 Filter Kombinasi

Fungsi filter kombinasi adalah menjalankan secara kombinasi filter median atau mean dalam kombinasi yang dipilih oleh pengguna. Kombinasi dapat secara acak, misalnya mean-median, atau median-mean, atau mean-mean-median, atau median-mean-median-median, dan seterusnya. Pengguna hanya menginputkan basis filter dalam urutan yang dikehendaki. Proses eksekusi dilakukan seperti cuplikan program utama dalam kode program di bawah ini.

Inputan kombinasi dari user disimpan dalam variabel *kombinasi* yang berbentuk string. Kemudian panjang string dibaca dan disimpan dalam variabel *max* (baris 1). *Max* berbentuk matriks berdimensi 1x2, panjang string disimpan dalam orde matriks *max(2)* (baris 4). Urutan basis filter adalah pada setiap karakter pada variabel kombinasi, misalnya urutan untuk mean-median-mean adalah 212. Maka pada saat iterasi pertama, *kombinasi(1)* bernilai 2, iterasi kedua, *kombinasi(2)* bernilai 1, dan *kombinasi(3)* bernilai 3. Judul disimpan dalam variabel *filt* yang berbentuk string, seperti teknik sebelumnya di pemrosesan filter rekursif (baris 7, 10, 12, dan 15). Untuk pilihan yang tidak valid (di luar basis 1 atau 2) maka ditampilkan string 'xx' dan tidak diproses filternya (baris 12).

```

1    max=size(kombinasi); %menghitung jumlah kombinasi
2    F=G; %memindahkan variable gambar bernoise ke F
3    filt='';
4    for i=1:max(2)
5        if kombinasi(i)=='1',
6            F=filter_median(F);

```

```

7         filt=[filt ' Median'];
8     elseif kombinasi(i) == '2',
9         F=filter_mean(F);
10        filt=[filt ' Mean'];
11    else
12        filt=[filt ' xx'];
13    end
14 end
15 filt=['Output Filter Kombinasi' filt];
16 fprintf('\n ');
17 fprintf(filt);
18 filt=filt;
19 figure();imshow(F);
20 title(filt);
21 psnr=psnr1(A,F);

```

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1. Implementasi

Tahapan implementasi merupakan tahapan dimana skrip program Matlab yang telah dianalisa dan ditulis, diuji untuk berbagai inputan citra, diukur performansinya serta diamati hasil keluarannya (citra output yang dihasilkan).

5.1.1. Lingkungan Implementasi

Komponen-komponen yang digunakan pada saat implementasi program noise filter ini adalah:

Perangkat keras

- a. Processor : *Intel(R) Centrino 1.50 GHz*
- b. Memori (RAM) : 1.0 GB

Perangkat Lunak

- a. Sistem Operasi : *Windows 7 Profesional (min Windows XP)*
- b. Bahasa Pemrograman : Matlab
- c. Spreadsheet : MS. Excel 2007

5.1.2. Skema Implementasi

Program utama diimplementasikan dalam bentuk skrip m-file Matlab, yang dijalankan pada consol Matlab. Setelah dieksekusi, proses yang terjadi sebagai berikut:

1. Proses input parameter: program meminta inputan user untuk citra pengujian, jenis dan intensitas noise, serta pemilihan filter.
2. Proses selanjutnya, program akan membaca file citra asli dan menyimpan ke dalam variabel, menampilkan citra asli yang bersih dari noise

3. Selanjutnya citra dikenai noise sesuai dengan jenis dan intensitas noise yang dipilih, tampilkan citra dan hitung PSNR-nya terhadap citra asli.
4. Puncaknya dilakukan proses filtering dengan memanggil fungsi filter yang telah dibuat. Tampilkan citra hasil filtering dan hitung PSNR-nya terhadap citra asli.

Sedangkan sumber citra yang digunakan untuk pengujian ini adalah citra standard Baboon (512 x 512 piksel) dan Lena (512 x 512 piksel), dan citra tidak standar Pas Foto (752 x 1052 piksel) seperti terlihat pada Gambar 5.1 berikut ini.



Gambar 5.1 Citra pengujian Baboon, Lena, dan Pas Foto

5.2. Pengujian dan Analisa Hasil Pengujian

Untuk pengujian keempat sistem filter yang dibuat dalam tugas akhir ini, dilakukan serangkaian skenario pengujian sebagai berikut:

1. Skenario 1: Noise Impulse (salt & pepper) dengan berbagai intensitas
 - Filter yang diuji : Median, Mean, Kombinasi Median Mean, Kombinasi Mean Median, Rekursif Mean, Rekursif Median
 - Intensitas Noise : 10% 20%, 30%, 40%, 50%
2. Skenario 2: Noise Additif (gaussian) dengan berbagai level standar deviasi
 - Filter yang diuji : Median, Mean, Kombinasi Median Mean, Kombinasi Mean Median, Rekursif Mean, Rekursif Median
 - Standard deviasi Noise : 10, 20, 30, 40, 50 dari maksimal 255

3. Skenario 3: Noise impulse (salt & pepper) dengan Filter Rekursif
 - Filter yang diuji : Rekursif Mean (1,2,3,4 iterasi), Rekursif Median (1,2,3,4 iterasi)
 - Intensitas Noise : 40%
4. Skenario 4: Noise Additif (gaussian) dengan Filter Rekursif
 - Filter yang diuji : Rekursif Mean (1,2,3,4 iterasi), Rekursif Median (1,2,3,4 iterasi)
 - Standard deviasi Noise : 40 dari maksimal 255

Pengujian dilakukan terhadap citra standard “Baboon” (mewakili citra yang kaya warna) dan “Lena” (mewakili citra sepi atau warna tua), dan satu citra tidak standard dari koleksi pribadi, yaitu “pas foto”. Citra standard “Baboon” dan “Lena” diujikan terhadap Skenario 1 dan 2, sedangkan untuk Skenario 3 dan 4 hanya menggunakan citra “pas foto” saja, yang memiliki banyak area-area dengan warna homogen.

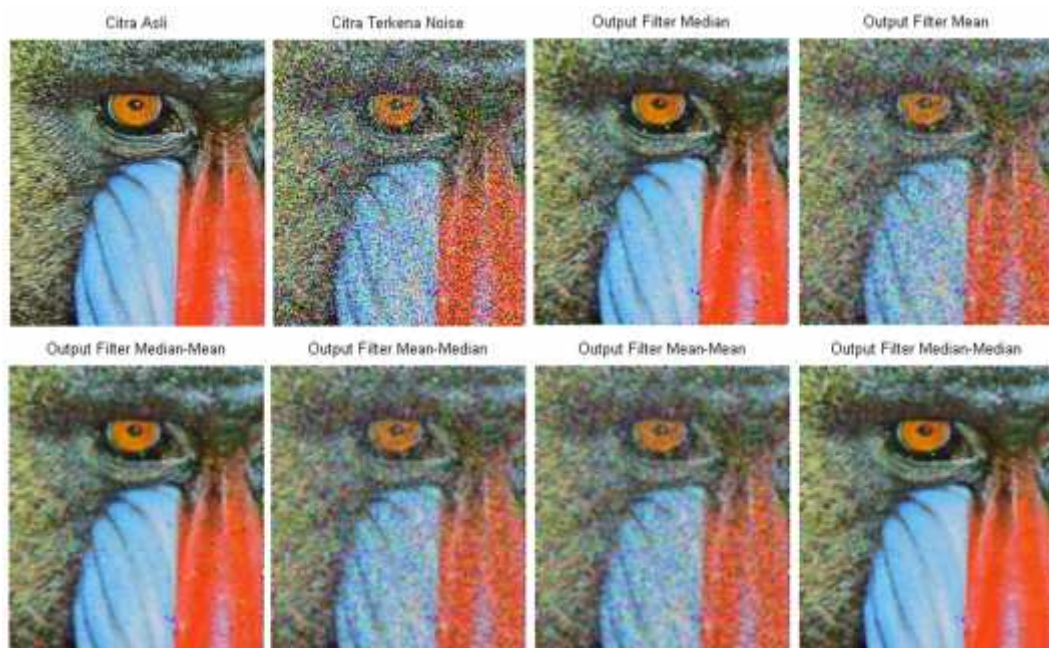
5.2.1. Data Pengujian

Skenario 1. Noise Impulse (salt & pepper) dengan berbagai intensitas

Hasil pengujian yang terukur dalam nilai PSNR dituliskan pada Tabel 5.1 dan 5.2 berikut ini, sedangkan tampilan citra untuk pengamatan kualitatif disajikan pada Gambar 5.2 dan 5.3 untuk citra Baboon dan Lena.

Tabel 5.1. Hasil Pengujian Skenario 1 untuk Impulse Noise citra Baboon

Citra \ Density	PSNR				
	10%	20%	30%	40%	50%
Citra ber-noise	15.25	12.25	10.5	9.27	8.28
output filter median	23.9	22.52	20.24	17.33	14.3
output filter mean	21.11	18.99	17.43	16.17	15.04
output filter median-mean	23.18	22.61	21.59	19.92	17.55
output filter mean-median	21.51	19.79	18.37	17.16	16.02
output filter mean-mean	21.5	19.85	18.44	17.24	16.1
output filter median-median	23.55	22.9	22.03	20.5	17.76



Gambar 5.2. Perbandingan citra-citra pengujian Skenario 1, impulse noise 30% (citra Baboon diperkecil)



Gambar 5.3. Perbandingan citra-citra pengujian Skenario 1 (Lena)

Tabel 5.2. Hasil Pengujian Skenario 1 untuk Impulse Noise citra Lena

Gambar \ Density	PSNR				
	10%	20%	30%	40%	50%
Citra ber-Noise	15.17	12.16	10.41	9.15	8.19
output filter median	32.16	28.29	23.23	18.59	15.04
output filter mean	23.46	20.2	18.12	16.47	15.21
output filter median-mean	30.94	29.14	26.48	22.8	19.32
ouut filter mean-median	24.92	21.67	19.5	17.71	16.33
output filter mean-mean	25.02	21.8	19.6	17.8	16.42
output filter median-median	31.8	29.91	27.85	24.12	19.74

Skenario 2. Gaussian noise dengan berbagai level standar deviasi

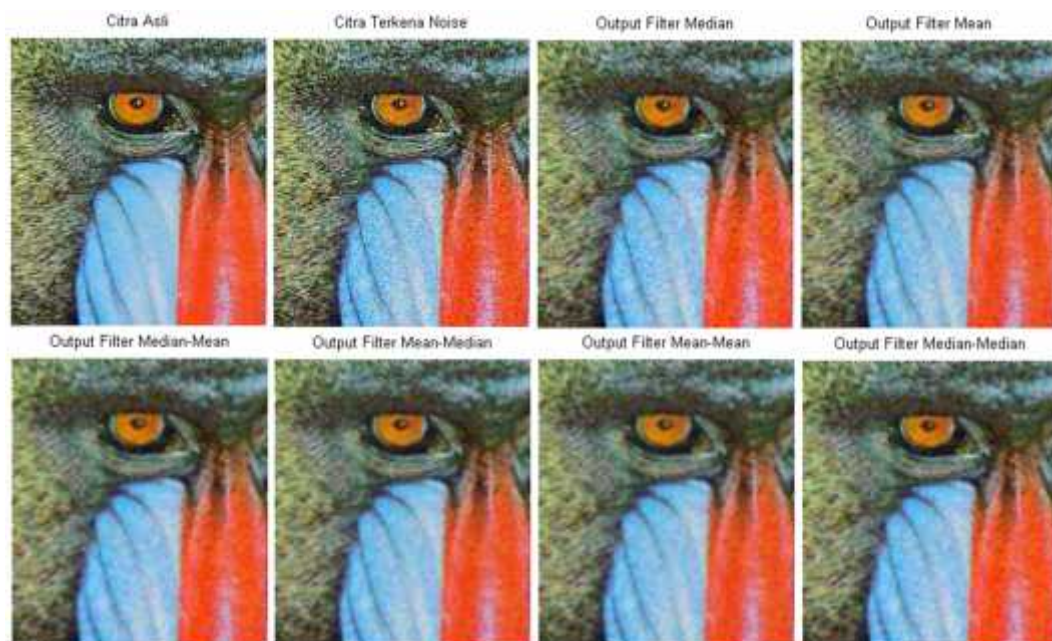
Hasil pengujian yang terukur dalam nilai PSNR dituliskan pada Tabel 5.3 dan 5.4 berikut ini, sedangkan tampilan citra untuk pengamatan kualitatif disajikan pada Gambar 5.4 dan 5.5 untuk citra Baboon dan Lena.

Tabel 5.3. Hasil Pengujian Skenario 2 untuk Gaussian Noise citra Baboon

Gambar \ Standard Deviasi	PSNR				
	10	20	30	40	50
Citra ber-Noise	28.16	22.22	18.8	16.46	14.74
output filter median	24.33	23.35	22.18	21.01	19.88
output filter mean	24.2	23.63	22.82	21.94	21.04
output filter median-mean	23.45	23.08	22.55	21.95	21.28
output filter mean-median	23.6	23.25	22.76	22.18	21.54
output filter mean-mean	23.35	23.09	22.67	22.16	21.58
output filter median-median	23.85	23.21	22.45	21.65	20.83

Tabel 5.4. Hasil Pengujian Skenario 2 untuk Gaussian Noise citra Lena

Gambar \ Standard Deviasi	PSNR				
	10	20	30	40	50
Citra ber-Noise	28.16	22.24	18.86	16.55	14.82
output filter median	32.42	28.46	25.62	23.43	21.66
output filter mean	32.6	29.61	27.08	25.05	23.39
output filter median-mean	31.72	29.7	27.76	26.03	24.51
output filter mean-median	32.18	30.2	28.26	26.51	25
output filter mean-mean	31.4	29.9	28.21	26.61	25.15
output filter median-median	32.36	29.46	27.13	25.21	23.58



Gambar 5.4. Perbandingan citra-citra pengujian Skenario 2 Gaussian noise $\sigma=30$ (citra Baboon)



Gambar 5.5. Perbandingan citra-citra pengujian Skenario 2, Gaussian noise $\sigma=30$ (citra Lena)

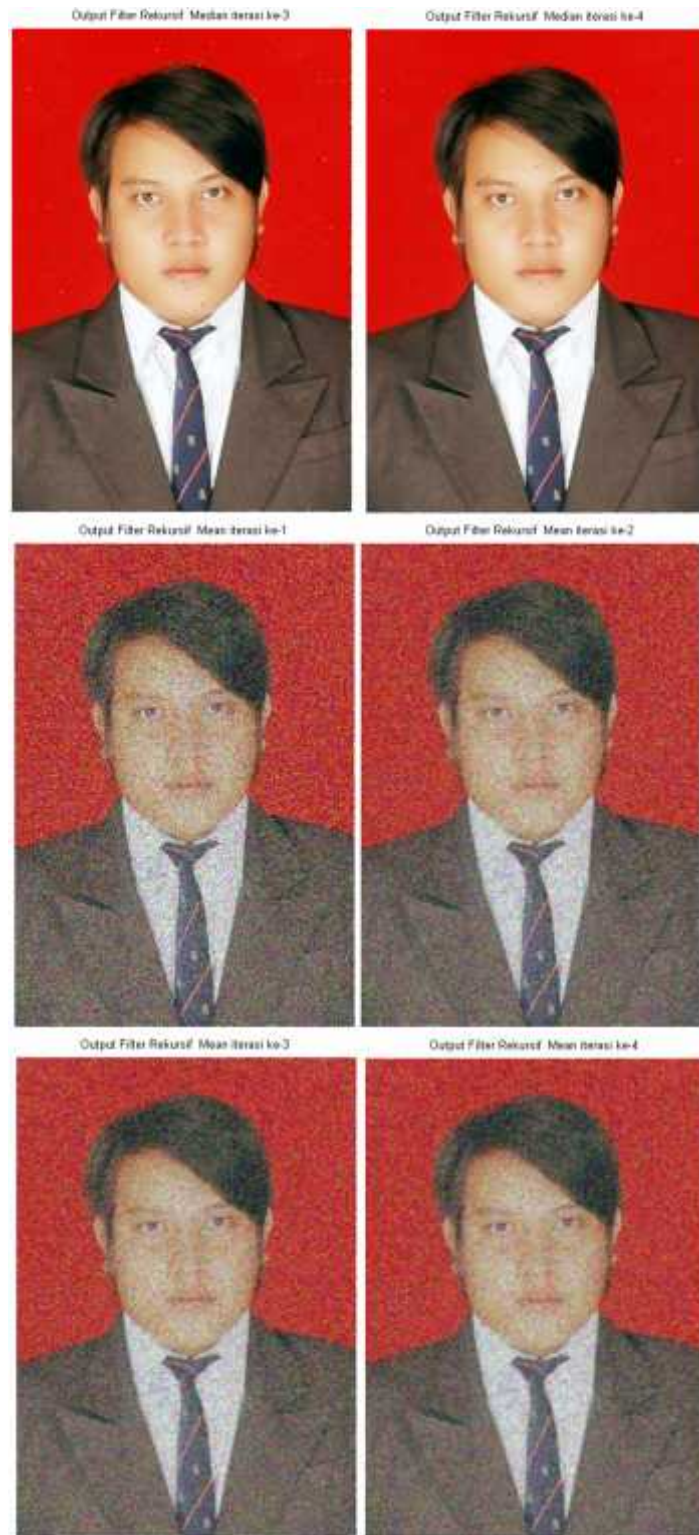
Skenario 3. Impulse noise dengan filter Rekursif pada citra "Pas Foto"

Hasil pengujian yang terukur dalam nilai PSNR dituliskan pada Tabel 5.5 berikut ini, sedangkan tampilan citra untuk pengamatan kualitatif disajikan pada Gambar 5.6 berikutnya.

Tabel 5.5. Hasil Pengujian Skenario 3 untuk Impulse Noise citra Pas Foto

Gambar \ Iterasi	PSNR			
	iter 1	iter 2	iter 3	iter 4
Citra Bernoise	8.36			
Base : Median	18.12	24.52	27.42	28.41
Base : Mean	14.78	15.79	16.15	16.34





Gambar 5.6. Perbandingan citra-citra pengujian Skenario 3 (Pas Foto) untuk Impulse Noise

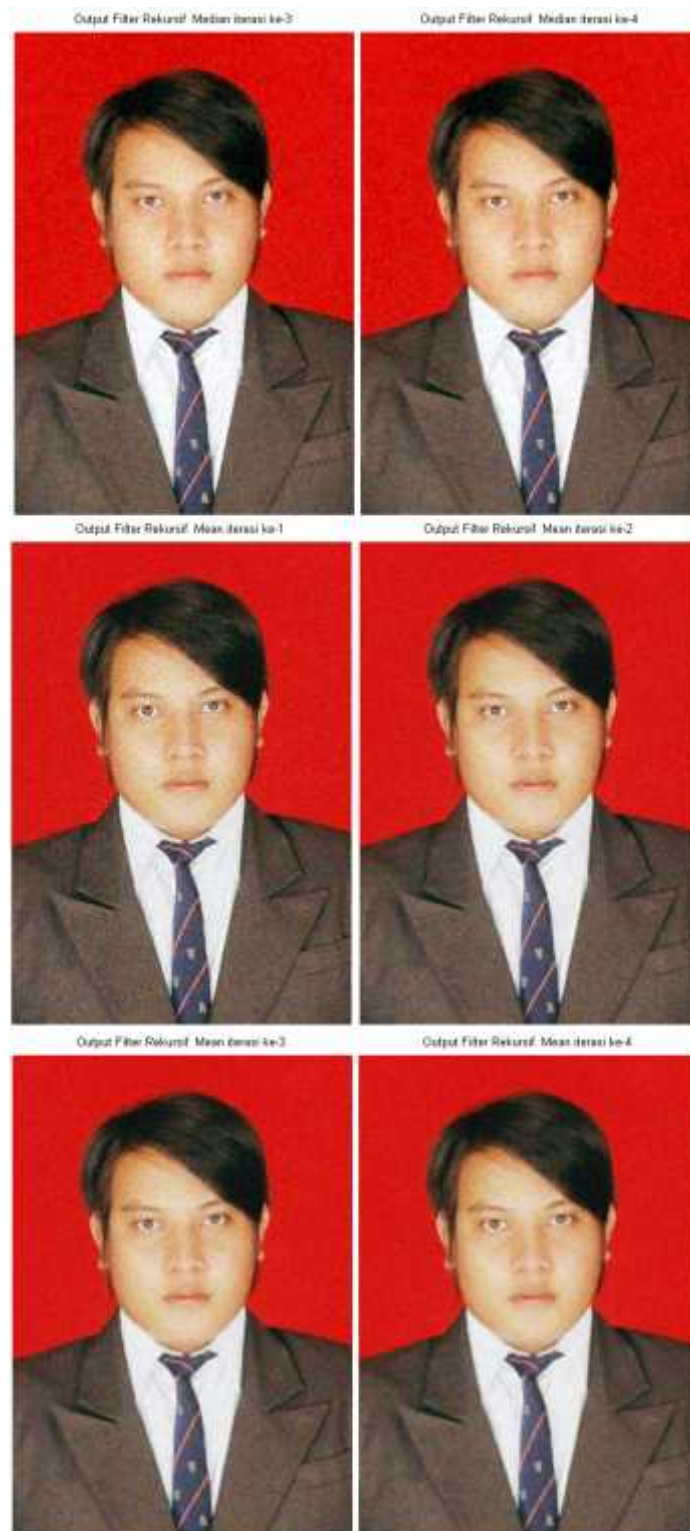
Skenario 4. Gaussian noise dengan filter Rekursif pada citra "Pas Foto"

Hasil pengujian yang terukur dalam nilai PSNR dituliskan pada Tabel 5.6 berikut ini, sedangkan tampilan citra untuk pengamatan kualitatif disajikan pada Gambar 5.7 berikutnya.

Tabel 5.6. Hasil Pengujian Skenario 4 untuk Gaussian Noise citra Pas Foto

Gambar \ Iterasi	PSNR			
	iter 1	iter 2	iter 3	iter 4
Citra Bernoise	17.28			
Base : Median	24.1	26.05	26.88	27.34
Base : Mean	25.12	26.56	26.94	27.07





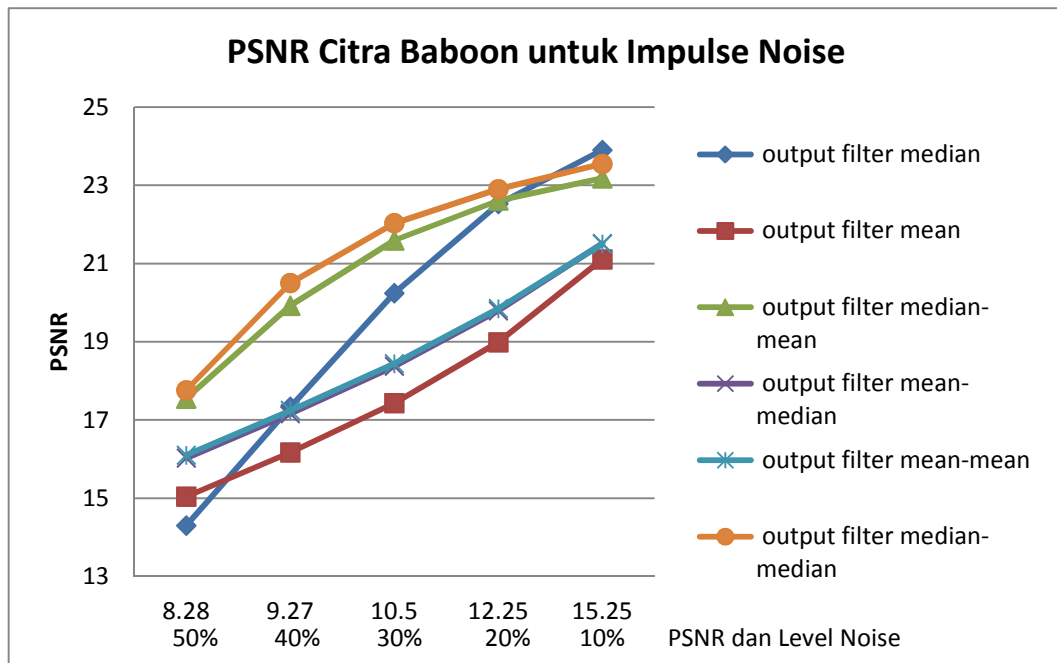
Gambar 5.7. Perbandingan citra-citra pengujian Skenario 4 (Pas Foto) untuk Gaussian Noise

5.2.2 Analisa Hasil Pengujian

Setelah dilakukan beberapa skenario pengujian seperti dijelaskan pada bagian sebelumnya, diperoleh hasil-hasil yang dapat diolah dan dianalisa sebagai berikut.

1. Skenario 1 citra Baboon

Untuk citra kaya warna Baboon, yang berukuran 512 x 512 piksel, pengenaaan impulse noise pada berbagai variasi intensitas menghasilkan grafik performansi PSNR seperti terlihat pada Gambar 5.8 berikut ini.



Gambar 5.8. PSNR dari output filter citra Baboon untuk Impulse Noise

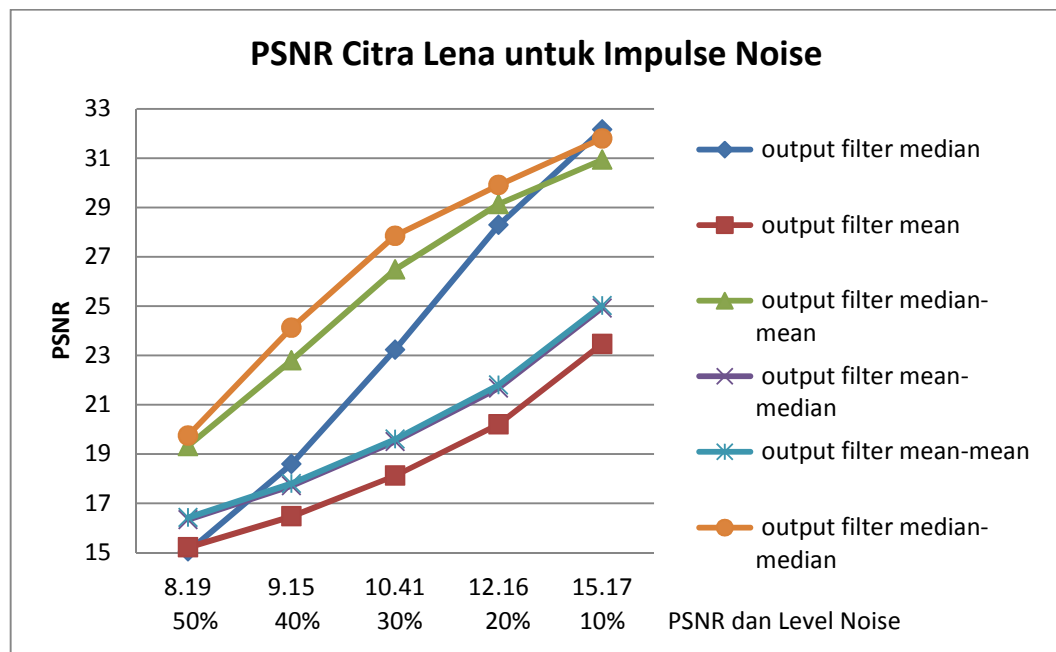
Dari grafik nilai PSNR di atas dapat dilihat bahwa performa yang terbaik untuk setiap level noise adalah filter rekursif Median-Median, yang terlihat dari posisinya yang selalu di atas. Pada noise yang rendah (PSNR noise bernilai 15.25), filter rekursif ini berada di bawah filter tunggal median nilai PSNR-nya, dan disusul dengan filter kombinasi Median-Mean. Hal ini masih normal karena level noise 10%, performa filter dengan PSNR di atas 23 sudah sangat bagus dalam menghilangkan noise. Dari pengamatan secara langsung terhadap citra pada Gambar 5.2, hal ini juga dapat dibuktikan, bahwa kualitas citra output dari filter

Median tunggal dan Median rekursif sangat bersih dibandingkan output citra filter lainnya. Dari Tabel 5.1 juga dapat dibandingkan angka PSNR-nya yang lebih tinggi (angka PSNR tertinggi diberi warna merah).

Dapat disimpulkan, untuk tipe noise Impulse, filter median, rekursif median maupun kombinasi dengan mendahulukan filter median akan memberikan hasil yang lebih baik.

2. Skenario 1 citra Lena

Untuk citra warna tua (sepia) Lena, yang berukuran 512 x 512 piksel, pengenalan impulse noise pada berbagai variasi intensitas menghasilkan grafik performansi PSNR seperti terlihat pada Gambar 5.9 berikut ini.



Gambar 5.9 PSNR dari output filter citra Lena untuk Impulse Noise

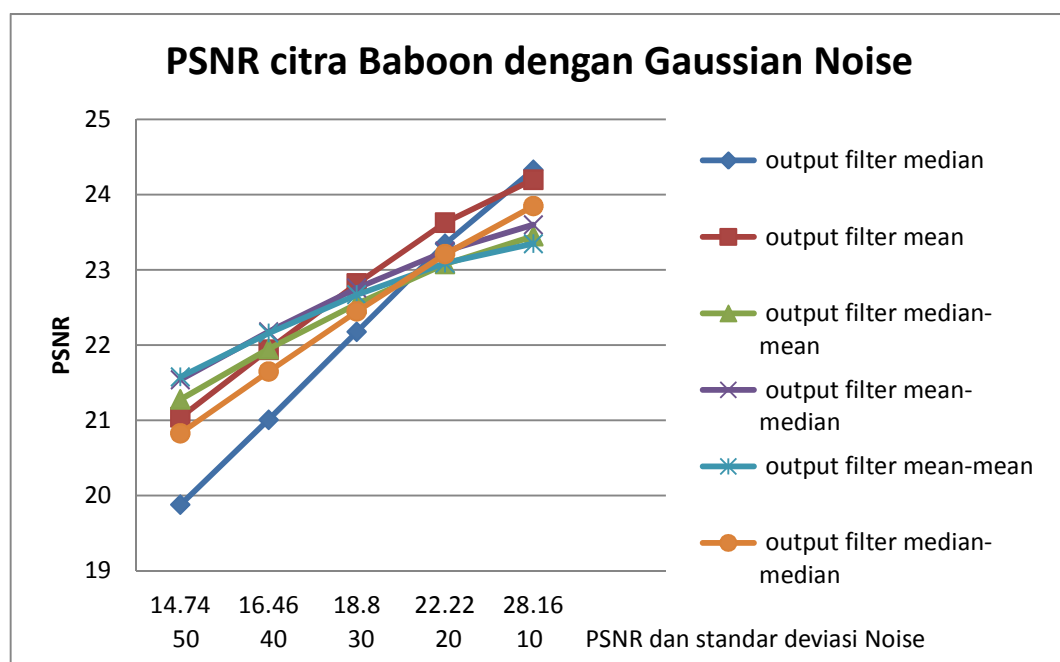
Dari grafik nilai PSNR untuk citra Lena di atas dapat dilihat bahwa performa yang terbaik secara keseluruhan adalah filter rekursif Median-Median, dan disusul oleh filter kombinasi Median-Mean. Sama seperti pada citra Baboon, pada intensitas noise yang rendah (PSNR noise bernilai 15.17), filter yang terbaik adalah filter Median tunggal. Ketiga filter: Median, Median-Median dan Median-

Mean secara PSNR sangat baik, karena berada di kisaran angka 31. Juga secara kualitas citra, output dari filter Median tunggal dan Median rekursif/kombinasi sangat bersih dibandingkan output citra filter lainnya, seperti ditunjukkan Gambar 5.3. Dari Tabel 5.2 juga dapat dibandingkan angka PSNR-nya yang lebih tinggi (angka PSNR tertinggi diberi warna merah).

Fakta ini semakin menguatkan kesimpulan yang pertama, yaitu, untuk tipe noise Impulse, filter median, rekursif median maupun kombinasi dengan mendahulukan filter median akan memberikan hasil yang lebih baik.

3. Skenario 2 citra Baboon

Untuk citra kaya warna Baboon, pengenaaan Gaussian noise pada berbagai level menghasilkan grafik performansi PSNR seperti Gambar 5.10 berikut ini.



Gambar 5.10 Grafik PSNR untuk Skenario 2 Citra Baboon

Dari grafik PSNR tersebut dapat diterangkan bahwa performa terbaik untuk level Gaussian noise tingkat tinggi ($\sigma=40$) adalah pada kombinasi dan rekursif dari Mean Filter. Namun performa pada level noise yang lebih rendah (PSNR 28.6) terbaik dicapai oleh Filter tunggal median dan mean. Hal ini

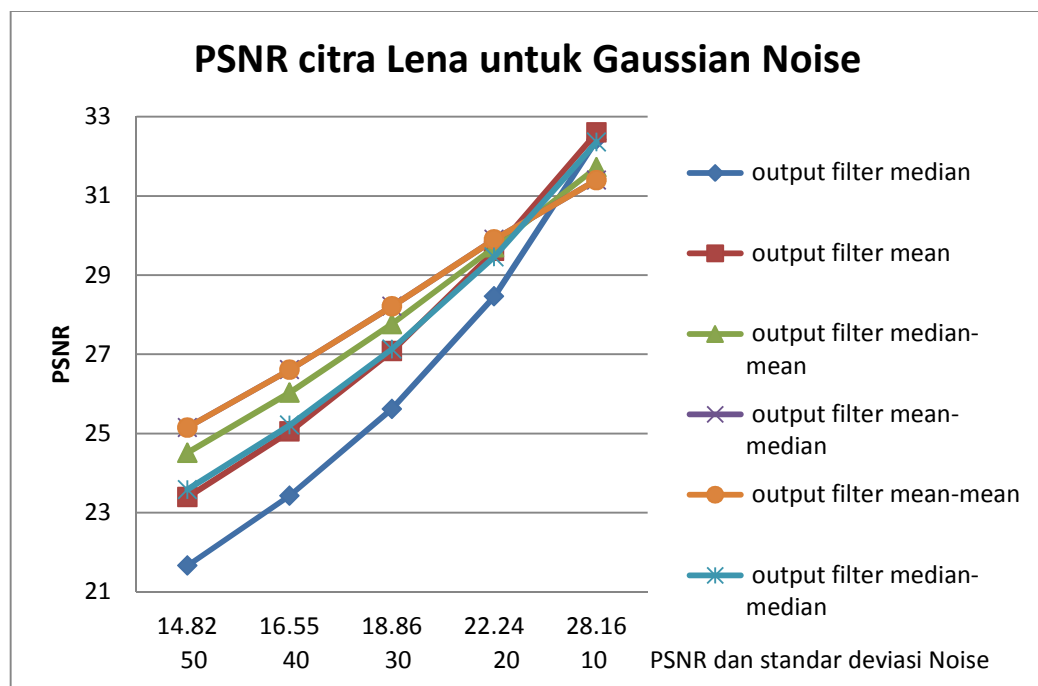
disebabkan karena level noise yang rendah, kualitas citra yang terkena noise masih belum terasa menurun, sehingga terkadang koreksi yang dilakukan oleh filter malah bisa menurunkan nilai PSNR. Secara kasat mata, noise yang rendah tidak terlalu berpengaruh terhadap penampakan citra, sehingga nilai PSNR filter yang lebih rendah dari PSNR noise juga tidak akan begitu terlihat efeknya.

Secara umum, filter yang memiliki performa cukup baik adalah Mean Filter, karena grafik PSNR –nya yang cenderung konstan menaik. Sedangkan secara kualitatif, pengamatan terhadap Gambar 5.4, yaitu perbandingan antara output masing-masing filter, secara umum performa Mean Filter memang cukup baik. Dari Tabel 5.3 juga terlihat bahwa filter mean dan rekursifnya memiliki angka PSNR yang lebih baik dibandingkan dengan filter median.

Dari hasil ini dapat disimpulkan bahwa basis filter mean cocok digunakan untuk mereduksi atau mengurangi noise berbentuk additive Gaussian.

4. Skenario 2 citra Lena

Untuk citra Lena, pengenaaan Gaussian noise pada berbagai variasi level menghasilkan grafik performansi PSNR seperti Gambar 5.11 berikut ini.



Gambar 5.11 Grafik PSNR untuk Skenario 2 Citra Lena

Secara umum grafik PSNR tersebut memperlihatkan bahwa performa terbaik untuk level Gaussian menggunakan rekursif atau kombinasi dari Mean Filter. Seperti citra Baboon, performa filter berbasis mean pada level noise yang lebih rendah (PSNR 28.16) terbaik dicapai oleh Filter tunggal. Secara kasat mata, noise Gaussian yang rendah tidak terlalu berpengaruh terhadap penampakan citra, sehingga nilai PSNR filter yang lebih rendah dari PSNR noise juga tidak akan begitu terlihat efeknya. Hal ini dapat dilihat secara cermat pada Gambar 5.4 dan 5.5 untuk citra Baboon dan Lena yang terkena Gaussian noise dengan standard deviasi 30.

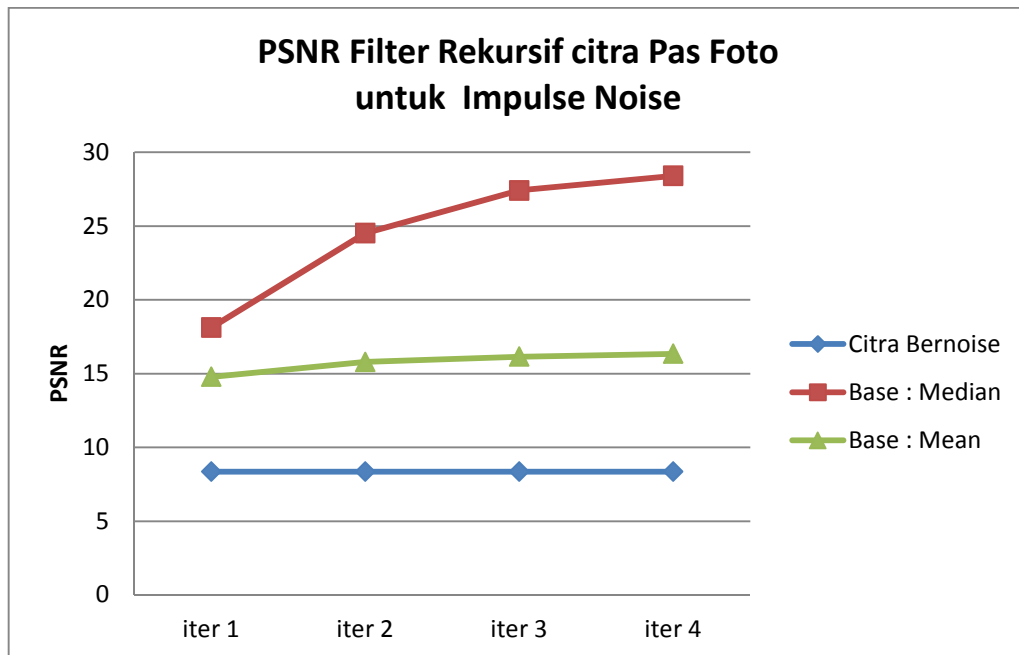
Secara kualitatif, pengamatan terhadap Gambar 5.5 (Lena), yaitu perbandingan antara output masing-masing filter, menunjukkan bahwa secara umum performa Mean Filter memang cukup baik. Dari Tabel 5.4 juga terlihat bahwa filter mean tunggal, rekursif dan kombinasinya memiliki performa yang lebih baik dibandingkan dengan filter lainnya.

Hasil ini semakin memperkuat kesimpulan kita bahwa basis filter mean cocok digunakan untuk mereduksi atau mengurangi noise berbentuk additive Gaussian.

5. Skenario 3 citra Pas Foto

Untuk citra Pas Foto, pengenalan Impulse noise pada intensitas tinggi 40% dengan filter rekursif 4x iterasi, menghasilkan grafik performansi PSNR filter rekursif basis median dan basis mean seperti terlihat pada Gambar 5.12 berikut ini.

Dari grafik pada Gambar 5.12 sangat terlihat bahwa filter median secara sukses menghilangkan noise impulse dengan intensitas yang tinggi (40%), yang ditunjukkan oleh peningkatan nilai PSNR secara signifikan. Sedangkan filter mean tidak terlalu baik dalam menghilangkan noise impulse tersebut walaupun sudah dilakukan secara rekursif, karena peningkatan nilai PSNR sangat kecil. Dari pengamatan kualitatif terhadap gambar 5.6, citra output setiap iterasi masih mengandung noise yang cukup mengganggu.



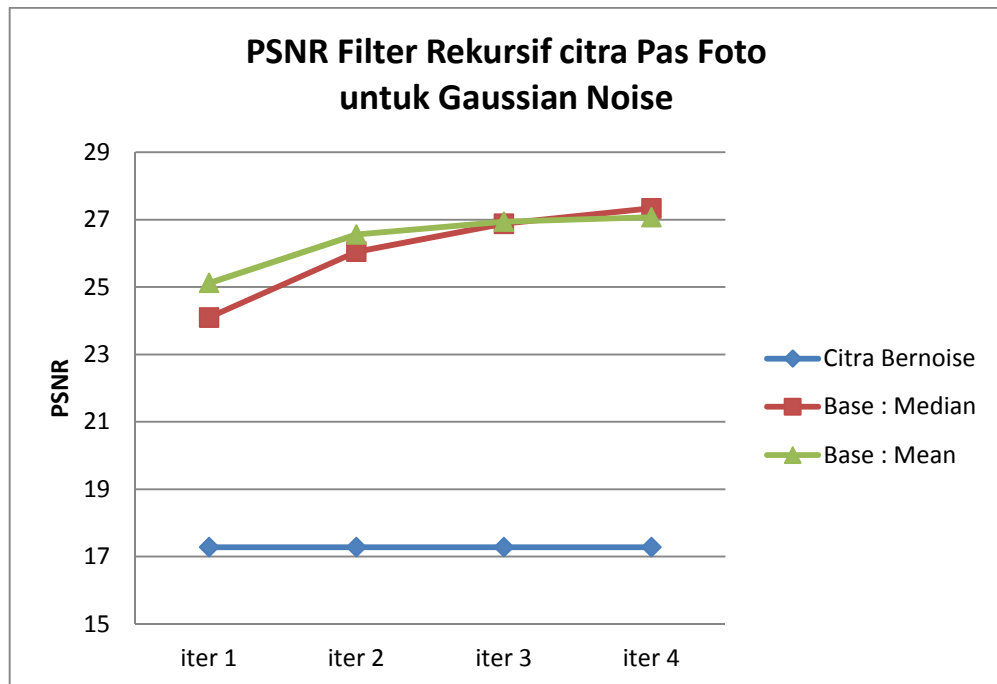
Gambar 5.12 Peningkatan nilai PSNR pada filter rekursif untuk impulse noise

Hal ini sekali lagi membuktikan bahwa untuk jenis noise berbentuk impulse, maka filter median sebagai basis sangat cocok digunakan pertama kali, baik secara tunggal maupun rekursif atau dikombinasikan dengan filter lainnya.

6. Skenario 3 citra Pas Foto

Untuk citra Pas Foto, pengenaaan Gaussian noise pada level standar deviasi tinggi ($\sigma=40$) dengan filter rekursif 4x iterasi, menghasilkan grafik performansi PSNR antara basis median dan basis mean seperti terlihat pada Gambar 5.13 berikut ini.

Dari grafik pada Gambar 5.13 di bawah, dapat dilihat bahwa filter rekursif mean dapat memperbaiki kualitas citra secara rata-rata lebih baik daripada filter median. Demikian pula secara kualitatif, dengan mengamati citra output dari masing-masing iterasi filter, filter mean menghasilkan gambar yang lebih baik dari filter median.



Gambar 5.13 Peningkatan nilai PSNR filter rekursif untuk Gaussian Noise

Walaupun pada iterasi yang lebih tinggi, nilai PSNR dari citra output median lebih tinggi, namun pada pengamatan kualitas citra seperti pada Gambar 5.7, citra output yang dihasilkan masih lebih baik filter mean.

5.2.3 Kesimpulan Hasil Pengujian

Setelah dilakukan beberapa skenario pengujian dan melakukan analisa terhadap hasilnya, dapat ditarik kesimpulan sebagai berikut:

1. Untuk citra kaya warna seperti Baboon dan citra sepia seperti Lena, pengenaan impulse noise dapat diatasi dengan menggunakan Median Filter. Semakin tinggi intensitas noise, penggunaan filter rekursif berbasis median filter akan memberikan hasil yang lebih baik.

Hal ini ditunjukkan oleh pengamatan kualitatif terhadap output filter pada Gambar 5.2 dan 5.3 (bagian output filter median-median dan output filter median). Dan juga ditunjukkan dengan lebih detail untuk filter rekursif di Gambar 5.7.

Secara kuantitatif, nilai PSNR dari filter-filter tersebut juga menempati posisi tertinggi, seperti ditunjukkan pada Tabel 5.1 dan 5.2, atau pada grafik PSNR di dalam Gambar 5.8 dan 5.9.

2. Pengenaan Gaussian noise untuk citra kaya warna maupun untuk citra sepia dapat diatasi dengan menggunakan mean filter. Hal ini sesuai dengan karakteristik noise yang terdistribusi secara gauss, di mana bila pusat window terkena noise dengan intensitas maksimum, maka di sekitar piksel pusat window tersebut juga akan terpengaruh noise dengan intensitas yang lebih rendah, sesuai dengan fungsi distribusi gauss.

Hal ini dapat dilihat pada hasil pengamatan kualitatif terhadap citra output filter, seperti Gambar 5.4 dan 5.5, yaitu bagian output filter mean, dan output filter mean-mean (rekursif). Secara detail, filter rekursif mean lebih baik dalam menghilangkan noise Gaussian ditunjukkan dalam Gambar 5.7.

Secara kuantitatif, nilai PSNR dari filter-filter tersebut juga menempati posisi yang lebih tinggi di banding filter median, seperti pada Tabel 5.3 dan Tabel 5.4, yang juga diperlihatkan secara grafik pada gambar 5.10 dan 5.11.

3. Penggunaan filter rekursif dapat menekan dan mereduksi noise secara signifikan untuk noise dengan level atau intensitas yang tinggi. Noise impulse dengan menggunakan rekursif median, dan noise Gaussian dengan menggunakan rekursif mean.

Hal ini ditunjukkan secara kualitatif dengan mengamati output filter pada gambar 5.6 untuk noise impulse, dan gambar 5.7 untuk noise Gaussian.

Secara kuantitatif, nilai PSNR yang meningkat ditunjukkan oleh Tabel 5.5 dan 5.6 seiring dengan naiknya iterasi, dan diperlihatkan pula pada grafik PSNR pada Gambar 5.12 dan 5.13.

4. Filter kombinasi juga baik untuk mengurangi noise dengan level atau intensitas yang tinggi. Namun demikian, mana filter pertama yang digunakan, tetap tergantung kepada tipe noise yang ada. Bila noise impulse, filter pertama kombinasi haruslah median filter, sedangkan bila noise Gaussian, filter pertama kombinasi haruslah mean filter.

Hal ini ditunjukkan oleh performa yang baik dari pengamatan kualitatif pada Gambar 5.2 dan 5.3 untuk tipe noise impulse, dan gambar 5.4-5.5 untuk tipe noise Gaussian.

Secara kuantitatif, terlihat bahwa filter kombinasi dengan urutan pertama yang sesuai dengan kesimpulan 1 dan 2 juga memberikan nilai PSNR yang lebih tinggi. Hal ini karena filter pertama sangat berperan dalam mengurangi noise secara signifikan.

BAB VI

PENUTUP

6.1 Kesimpulan

Kesimpulan yang dapat diambil dari pengolahan citra untuk pengurangan noise dengan menggunakan metode Median Filter, Mean Filter, Filter Rekursif dan Kombinasi, adalah sebagai berikut:

1. Untuk mengurangi impulse noise, filter yang terbaik adalah Median Filter. Semakin tinggi intensitas noise, penggunaan filter Median secara rekursif maupun kombinasi dengan filter lain akan memberikan hasil yang lebih baik.
2. Untuk mengurangi Gaussian noise, filter yang lebih baik adalah Mean Filter. Semakin tinggi level noise, penggunaan filter Mean secara rekursif maupun kombinasi dengan filter lain akan memberikan hasil yang lebih baik.
3. Filter Mean dan Median, secara tunggal maupun rekursif atau kombinasi dapat bekerja secara efektif untuk mengurangi noise pada citra kaya warna maupun citra sepia bila diterapkan secara tepat sesuai dengan jenis noise yang dihadapi.
4. Hasil pengamatan secara kualitatif, median filter terbukti mampu membersihkan citra dari noise impulse, dan mean filter dapat mengurangi Gaussian noise secara signifikan. Sedangkan secara kuantitatif, hal ini juga ditunjukkan oleh nilai PSNR yang lebih tinggi yang menyatakan performa masing-masing filter dalam mereduksi noise.

6.2 Saran

Beberapa hal yang dapat disarankan dalam penelitian selanjutnya adalah:

1. Untuk proses mendeteksi dan mereduksi noise gaussian yang lebih sulit, perlu dikembangkan lagi atau dimodifikasi metode filter berbasis mean, misalnya menggunakan fuzzy mean.
2. Mengembangkan metode lainnya yang tidak berbasis pada statistik untuk mereduksi noise.

DAFTAR PUSTAKA

- Bambang Yuwono (2010), *Image Smoothing Menggunakan Mean Filtering, Median Filtering, Modus Filtering, dan Gaussian Filtering*, Jurnal Telematika, Vol. 7, No. 1, Juli 2010, hlm 65-75.
- Church, J.C., Y. Chen, and S.V. Rice (2008), *A Spatial Medial Filter for Noise Removal in Digital Images*, Proc. Of IEEE Southeast Conference (SECON), pp. 618-623, Huntsville, Alabama, April 2008.
- Eko Prasetyo (2011), *Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab*, Penerbit Andi, ISBN: 978-979-29-2703-0, Yogyakarta, 2011.
- Gupta, Gajanand (2011), *Algorithm for image Processing Using Improved Median Filter and Comparison of Mean, Median and Improved Median Filter*, International Journal of Soft Computing and Engineering (IJSCE) ISSN : 2231-2307, Volume-1, Issue-5, November 2011, Hlm. 304-311
- Gonzalez, Rafael C., Richard E. Woods (2008), *Digital Image Processing Third Edition*, Pearson Prentice Hall, Upper Saddle River, New Jersey
- Hwang, H and R.A Haddad (1995), *Adaptive Median Filters : New Algorithm and Result*, IEEE Transactions on Image Processing, Vol 4, No.4, April 1995 Hlm 499-502.
- Murinto, Eko Aribowo, Risnadi Syazali (2007) *Analisis Perbandingan Metode Intensity Filtering Dengan Metode Frequency Filtering Sebagai Reduksi Noise Pada Citra Digital*, Seminar Nasional Aplikasi Teknologi Informasi 2007 (SNATI 2007) ISSN: 1907-5022, Yogyakarta, 2007
- Patidar, Pawan, Manoj Gupta, Sumit Srivastava, Ashok Kumar Nagawat (2010), *Image De-noising by Various Filters for Different Noise*, International Journal of Computer Applications, ISSN : 0975-8887 Volume 9 – No.4, November 2010, Hlm.45-50
- Shinde, Bhausheb, Dnyandeo Mhaske, Machindra Patare, A.R. Dani (2012), *Apply Different Filtering Techniquet to Remove The Speckle Noise Using Medical Images*. International Journal of Engineering Research and Application (IJERA) ISSN : 2248-9622 Vol.2, Issue 1, Jan – Feb 2012, pp.1071-1079

Silvester Tena (2009), *Image Enhancement Menggunakan Metode Linear Filtering dan Stationary Wavelet Transform*, Jurnal Teknologi Elektro Vol. 32 No.2 Juli - Desember 2009

Tuti Adi Prihatin (2010), *Analisis dan Implementasi Low Pass Filter untuk Mereduksi Noise pada Citra Digital*, Skripsi, Universitas Sumatera Utara, 2010